102

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Attorney Docket No. 006335.00007

In the Application of:	)	
	)	
George M. Levinson, et al.	)	Examiner: TSAI, Carlos S. W.
	)	
Serial No.: 10/747,974	)	
	)	Group Art Unit: 2857
Filed: December 30, 2003	)	
	)	
For: System, Method, and Computer-Readable Medium For Collection of Environment Data and Generation of User Report For Compliance With FDA Requirements	)	

DECLARATION OF GEORGE M. LEVINSON PURSUANT TO RULE 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

I, George M. Levinson, declare as follows:

1. I am a co-inventor of the subject matter disclosed and claimed in the above-cited application.

2. I have read the Office Action mailed August 30, 2004 in the above-cited application, and the Response thereto, and I understand that this Declaration will be filed with the Response.

3. My co-inventors, Timothy P. Joyce, Victoria Galliani, and I conceived of the subject matter disclosed and claimed in the above-cited application at least as early as July 22, 2002. See attached Exhibit A, Requirements Definition for EMSS Version 3.0. This document states, among other things, that the "system shall provide a method of assessing and documenting information

Banner & Witcoff, Ltd.
10 South Wacker Drive
Suite 3000
Chicago, Illinois 60606
Phone: (312) 463-5000

associated with Environmental monitoring including, but not limited to, collecting microbial counts, analyzing, and trending of entered data."

4. My co-inventors, Timothy P. Joyce, Victoria Galliani, and I were reasonably diligent in reducing the invention to practice prior to June 19, 2003. For example, on September 4, 1992, we prepared a "Functional Specification for EMSS Version 3.0." See attached Exhibit B, which describes in further detail what the system will do.

5. Further, on January 3, 2003, Timothy P. Joyce, and an employee (Tony Castellano) of the assignee of our invention (Compliance Software Solutions Corp.) under Mr. Joyce's direction, prepared a "Functional Specification," a "Requirements Definition," and a "Configuration Management Plan." See attached Exhibit C. The "Functional Specification" noted, among other things, that the "system shall provide a method of importing/transferring data output from the Becton Dickinson Phoenix™ Automated Microbiology System into the Compliance Software Solutions Corp Environmental Monitoring Software System using a CFR Part 11 compliant data management environment." As also noted in Exhibit C, "The Becton Dickinson Phoenix™ Automated Microbiology System is intended for rapid identification (ID) and antimicrobial susceptibility testing (AST) of significant organisms." Exhibit C also sets forth the "User Requirements/Functional Specifications" of the "interface."

6. At least as early as March 7, 2003, Timothy P. Joyce prepared a system diagram showing the system having a "converter" as envisioned by the inventors. See Exhibit D, including the computer screen shot showing the date of the diagram as March 7, 2003.

Banner & Witcoff, Ltd.
10 South Wacker Drive
Suite 3000
Chicago, Illinois 60606
Phone: (312) 463-5000

7. Mr. Joyce also provided specifications for a port array to a vendor (Millennium Electronics, Inc.), and instructed the vendor to prepare a "Network Serial Port Array Software Interface Specification" dated June 12, 2003, and prototype that would create an interface between Becton Dickinson Phoenix™ Automated Microbiology System (see http://www.difco.com/) and the system we developed in accordance with our invention. See attached Exhibit E. Under Mr. Joyce's direction, an employee (Tony Castellano) of the assignee of our invention (Compliance Software Solutions Corp.) prepared two separate Revision Summaries on May 9, 2003. See attached Exhibits F and G. In Exhibit F, reference is made to the interface between a Becton Dickinson Phoenix™ Automated Microbiology System and the system we developed in accordance with our invention. In Exhibit G, a further description is made regarding the database objects, and includes further reference to the interface between a Becton Dickinson Phoenix™ Automated Microbiology System and the system we developed in accordance with our invention.

8. I contacted counsel regarding the invention on July 17, 2003. Thereafter, Timothy P. Joyce and I reviewed patent search results and drafts of the application up until the filing of the application on December 30, 2003.

9. The "port array" or "interface" or "converter" identified in the attached Exhibits is referred to as the "universal hub" in the pending patent claims.

10. All of the above activities took place within the United States.

11. I have been warned that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and may jeopardize the validity of the application or any

patent issuing thereon. All statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

Respectfully submitted,

Dated: December 29, 2004

George M. Levinson

EXHIBIT A

Approvals:

T. Joyce — Date: 7-22-02

G. Levinson, Marketing — Date: 7/22/02

V. Galliani, Quality Systems — Date: 7/22/02

Overview

The system shall provide a method of assessing and documenting information associated with Environmental Monitoring including, but not limited to, collecting microbial counts, analyzing, and trending of entered data. These trends provide valuable insight into the effectiveness of decontamination procedures, housekeeping practices, personnel training, and the potential for microbial build-up during production. The use of the system will help ensure that environmental control systems are operating as intended.

Features

SYSTEM REQUIREMENTS

The software is designed to reside on a local workstation and its database on a network server to: 1) take advantage of the out-of-specification e-mail notification using existing mail network, and 2) provide multiple workstations access to the same system database. The software can function without a network (i.e. with the database and application loaded on a local workstation).

The equipment recommendations are as follows:

Workstation:

- Pentium or any 32 bit microprocessor – minimum speed 400 MHz
- Microsoft Windows 98 Second Edition or higher
- 64 MB RAM (minimum) / 128 MB RAM recommended

Server:

- Minimum Pentium III
- 256 MB RAM (minimum) / 1.0 GB RAM recommended
- MSDE, SQL Server 7.0 or higher, or Oracle 8x or higher installed

USER REQUIREMENTS

The system should should, at a minimum:

Provide adequate security for the data
Allow for user-defined security roles to manage user's access to system functions / data
Allow system users to easily manage the configuration of the areas to be monitored
Allow for recording of all pertinent information for sampling materials and equipment
Provide a means of recording all pertinent information for sampling sites, including sample identification, dates / times, etc.
Allow for the management of data according to potential sample incubation / exposure periods
Allow for recording of organism identification data
Allow for a means of labeling sample containers
Provide a means of managing the sampling schedule / task lists by area, personnel, etc.
Provide a means of managing 'second-party' approval / review of recorded data
Provide 'user-friendly' / efficient methods for recording of all system data
Provide a means of tracking all modifications to system data (change control / audit)
Utilize a means of analyzing entered data to notify appropriate personnel upon detection of a condition warranting such action
Provide a means of managing / tracking out-of-specification results from event occurrence through conclusion of investigation
Ensure the data management process is fully CFR part 11 compliant
Provide a comprehensive reporting package for final results allowing users to select graph / plot style and query parameters
Provide a comprehensive reporting package for final results allowing users to select report format and query parameters
Allow for potential 'transfer / export' of managed data to other data formats

EXHIBIT B

Approvals:

T. Joyce, 9-3-02 Date

G. Levinson, Marketing 9-4-02 Date

V. Galliani, Quality Systems 9-4-02 Date

Overview

- Application Security / Management
  - User access
  - System defaults
  - Application 'preset' parameters
- Database Configuration / Management
  - Plant
  - Building
  - Area
  - Location
  - Sample Type
  - Group / Worklist
- Data Processing
  - Sample taking
  - Initial Data recording
  - Final Data processing
    - Out-of-specification results (including deviation management)
    - OK results
    - Isolate identification
  - Data review / approval
  - Data modification
- Data Output / Reporting
  - User-configurable report (query) parameters
  - User-configurable report (plot) formats
  - 'Standard' (non user-configurable) reports
  - Process utilities (worksheets, labels, etc.)
  - Electronic output
    - E-mail notifications
      - Out-of-specification results
      - 'Late / missing' samples, groups / worklists, etc.
  - Export options (i.e. PDF, etc.)

Potential User Scenarios

Scenario 1: Jon Fishman – Jon is a microbiology technician in your basic pharmaceutical manufacturing facility's Quality Control department. Jon doesn't make a whole lot of money and is usually burdened with the most menial tasks involving environmental monitoring, including taking and identifying the samples (all the while using supplies/equipment that is within expiration), ensuring the correct samples/sampling groups are monitored according to their scheduled days / processes, incubating the samples, keeping track of the incubation periods of stacks of plates (so they are incubated for the correct period of time), counting and recording results, and notifying his supervisor when "bad things go happen." Jon also is responsible for recording all information associated with the environmental monitoring process for the big bad FDA inspector each time monitoring is performed, including sample dates/times, media lots, equipment IDs, sample IDs, etc. Jon hates writing on data sheets/in laboratory notebooks and would love an easier way to do his job without disrupting his routine or making him work harder (with no more pay!) – automating the management of some of Jon's tasks would make him very happy, but he has rudimentary technological skills (and that's being kind). He doesn't like to type either.

Scenario 2: Mike Gordon – Mike is Jon's direct supervisor, and manages five other microbiology technicians in the Quality Control department. Mike determines the environmental monitoring schedule, which areas are monitored and when/by whom, and usually delegates the monitoring tasks among his six microbiology technicians. Mike is responsible for reviewing the data that is processed/recorded by Jon and the rest of the microbiology technicians for completeness/correctness. Mike is also notified when something goes wrong with the monitoring process – this includes out-of-specification results, missing/incorrect sampling data that needs to be changed, etc. When "bad things go happen," it is Mike's responsibility to investigate/interrogate the situation to determine what happened, its impact regarding the manufacturing processes associated with the environmental monitoring that has "gone awry," determine which actions, if any, need to be taken as a result of the event, and document the conclusion/closure of the excursion. Mike is in frequent contact with the operations managers, both to find out what is going on in the manufacturing areas (and thus what monitoring needs to be done on which days) and to keep them up to speed on the results of "good" and "bad" monitoring data. Mike spends a lot of time on the phone with both the operations staff, outside vendors (for ordering environmental monitoring supplies), and his boss (Director / Associate Director QC), who always wants a "nice, tidy, concise snapshot" of what was going on in manufacturing room A during week B while product C was being made. Mike is semi-skilled with technology (i.e. he can use charts in Microsoft Excel, even though it takes an ENORMOUS amount of time from his day), but isn't afraid of it if it will give him more time for his everyday tasks (he doesn't like change much either) and make his life easier in general.

Potential User Scenarios (cont.)

Scenario 3: Page McConnell – Page is the Director of Quality Control and is Mike's boss. Page is the guy the Board of Directors, VP's, the "money men" look to when the bottom line doesn't look as good as they (and their free-spending trophy wives) would hope. Page runs the entire QC department – this includes requesting/approving all documentation generated by Mike, such as the out-of-specification reports and the "nice, tidy, concise snapshot" reports displaying the wonderfully tight grasp Page has on the entire manufacturing facility regarding cleanliness. Page likes nothing more than a report which looks like it has been overrun with zeroes. In fact, zero is Page's favorite number because the more zeroes he can throw up in his environmental monitoring data presentations at monthly board meetings, the less the "money men" have to spend on attorneys to protect the company against product recall lawsuits, insurance policies, etc. (and thus the more carats in their wives' gold-plated Range Rovers). This all makes Page look good, and a potential future candidate for "money man" status during the trek up the corporate ladder. Page wouldn't mind helping Mike out with tasks like creating the "nice, tidy, concise" reports because, of course, Page was promoted from Mike's position not too long ago and knows how hectic the Supervisor's life (or lack thereof) can be. Unfortunately, Page's technological savvy pretty much tops out at sending/receiving email (no joke – Page often has problems with the concepts of 'Reply' versus 'Reply All'), so the simpler the process of creating these "nice, tidy, concise" reports, the more likely Page will start to dabble himself instead of bothering Mike. After all, Page is the person the big, bad FDA inspector spends all his time with during site visits, and Page would like nothing better than to be able to produce data for Mr. FDA as quickly as possible (instead of going through Mike) – Page wants to get that inspector off the site before he finds the problems with the manufacturing facility that have been buried behind years of data records.

Scenario 4: Trey Anastasio – Trey is the IT manager, and the majority of Trey's day is taken up by answering stupid questions from the myriad of users somehow gainfully employed at the manufacturing facility because his staff consists of himself most days. When Trey isn't teaching somebody the difference between a file and a folder over the phone, he's roaming the site, correcting network settings, updating registry entries, etc., not to mention installing the latest "bells and whistles" associated with some software a "money man" read about in this week's Fortune magazine. The last thing in the world Trey likes to hear is that a corporate directive is coming down about a new software application that will make the Quality Control department more productive – not that Trey doesn't like or trust new software (he is in IT), but he doesn't need to be responsible for the implementation, maintenance, and support of another corporate/department-wide application. Trey is very protective of his "equipment" (not that equipment) and, like any good IT guy, likes to keep his network, servers, etc. exactly the same once they are working correctly. Unfortunately, the "money men" often make these decisions without consulting Trey, and, even though he's a good guy, Trey can be very reluctant to assist in the deployment of any new applications (to be honest, Trey gets his feelings hurt pretty easily when not consulted by the "bigwigs" and he feels like little more than a technical support guy at a small ISP than a guy with 42 different Microsoft certifications).

Non-Goals

Non-Goal 1: Automated Scheduler – The proposed system will not act as a scheduler for managing an existing environmental monitoring program. The system will not automatically (i.e. without user input) begin the data management process according to schedules defined within the monitoring program. This is not to say, however, that the system won't 'oversee' the daily use/tasks performed by the users and potentially notifying the appropriate personnel if a scheduled task has not been started (envision a 'watchdog' reminder system).

Non-Goal 2: Future Modules – The proposed system will not worry about "building in" interfaces for functional modules in the conceptual stage, such as managing Media Fill data, etc. We will rely on the proper design of the graphical user interface and, especially, the database to allow for retrofitting of such add-on functions in the future.

Non-Goal 3: Windows 95 Compatibility – The proposed system will not worry about installations/deployments on machines using Windows 95 as an OS. Anyplace that still has Windows 95 machines around has bigger fish to fry.

Initial Startup Parameters

Database Connection Parameters

- At initial startup, the user will be prompted for the database version (Oracle or SQL Server) and its corresponding database connection parameters

Default User Account Creation

- At initial startup (following the setting of database connection parameters) the user will be prompted to set up a 'SYSTEM' user account for access to the application

Application Security - Management

- Application Security / Management
  - User access – Accounts / account management
    - User accounts should be managed within the application (a separate security level from the database / OS security)
    - A 'synchronization' function between internal (application) security and external (database/OS) security should be included
    - User accounts (internal) will contain a userid and password which will be used to validate access to the application (application/GUI level security)
    - User accounts and their functional access within the application will be 'configurable' – i.e. using a list of application functions with checkboxes or some other selection method to specify which system functions each account will have access to
    - There will be an administrator role (pre-configured) which will be able to add, delete, and modify user account parameters
    - Administrators will have the ability to manually 'disable' user accounts
    - Passwords will be 'encrypted' within the system (non-viewable, even by administrator)
    - The password change period will be set by an administrator on an account-by-account basis (to accommodate temporary employees)
    - A password expiration date may be set by an administrator
    - New user accounts will require a change of password at first login
    - Passwords will have a minimum length of 6 characters and a maximum length of 15 characters
    - Passwords may contain a combination of alpha and numeric characters
    - There should be a system option forcing passwords to contain at least 1 alpha and 1 numeric character
    - Passwords will be case-specific
    - Userids must be unique
    - User account records will contain the full name manifestation (first and last name)
    - All users should be able to change his/her password at user prompting as well (i.e. via a 'Change Password' button) – in this case, the change period for this user account should be re-set to the 'beginning' – i.e. a User Account Management form
    - Upon password expiration, the user should be prompted to enter a new password
    - When changing passwords, the user should be prompted to enter old password, new password, and to re-enter new password to confirm password change

Application Security – Management (cont.)

- User access – Login
  - The system will require input of a valid userid / password before loading the GUI main screen
  - The system will autofill the login form with the userid from the last successful login
  - If an invalid userid and/or password is input, the system will notify the user with a message box that the userid and/or password are invalid
  - The system will disable a user account after three (3) unsuccessful login attempts
  - The system will notify the administrator upon disabling of an account
  - A disabled user account must be reinstated by the system administrator
  - The system will prevent the 'excising' (cut, copy, paste) of the login components (userid / password)

Environmental Monitoring Site Components

- The following components should be defined within the system, and each component will include the listed parameters:
  - Company / Site (user created)
    - Description (including address, city, state, zip)
  - Building (user created)
    - Description
    - Company / Site
  - Area (Room) (user created)
    - Description
    - Building
    - Classification - may have multiple classifications within an Area
    - Visual representation
  - Sample Site / Location (user created)
    - Description (Physical location)
    - Location reference number
    - Area
    - Classification
    - Site type
    - Sample type
    - Alert / action levels
  - Work / Task List (user created)
    - Description
    - Frequency
    - Sample locations / sample types to be monitored
  - Sample Type (user modified)*
    - Description
    - Incubation period
    - Visual representation
    - Precision (decimal places for results)
    - List of required fields (input formats)
    - Result entry options (special processing)
  - Classification (user modified)*
    - Description
  - Site Type (user modified)*
    - Description

* Denotes that the application will come 'pre-configured' with default components / values for Sample Type, Classification, and Site Type entries

Environmental Monitoring Site Components (cont.)

- EM Component Configuration
  - Localization options - At startup the program will prompt the user to review the default components for Sample Type, Classification, and Site Type for correspondence to local (i.e. client SOP) terminology
  - If at startup the program detects there are no EM components, the user will be guided through the process of creating components of the environmental monitoring site, including company / site, buildings, areas, sample locations
  - The application will allow for the user to modify the components post-initialization

Data Processing

Routine Environmental Monitoring

- The system should manage the following steps and their associated pieces of information of routine environmental monitoring:
  - Prompting the user for input of 'pre-monitoring' information such as media lot / equipment ID numbers, operation / process, product lot, etc.
  - Allow for monitoring of pre-defined environmental monitoring 'worklists,' individual sampling locations, or a combination of both
  - Creation of data sheets for recording initial / final environmental monitoring data
  - Creation of labels for identifying environmental monitoring sampling supplies (plates, etc.)
  - Provide a 'user-friendly' method for recording of all system data (data entry)
  - Allow for management / entry of 'initial' environmental monitoring data including sample date / time, sampler identification, etc.
  - Provide a check against entered 'initial' data for validity of data (i.e. data formatting, entry of invalid date/time with relation to the 'centralized' system clock, etc.)
  - Allow for variable incubation periods for environmental samples
  - Allow for variable exposure periods for environmental samples
  - Allow for management / entry of 'final' environmental monitoring data including read date / time, reader identification, final count, etc.
  - Provide a check against entered 'final' data for validity of data (i.e. data formatting, entry of invalid date/time with relation to the 'centralized' system clock, etc.)
  - Provide a means of analyzing entered results (final counts) versus defined specifications for that sample location at the point of data entry
  - Upon detection of an out-of-specification condition regarding entered results, notify appropriate personnel of the aberrant event
  - Provide a means of managing / tracking such out-of-specification conditions from event occurrence to conclusion / closure of the event
  - Notifying the appropriate management-level personnel whether 'scheduled' monitoring has taken place or is in process upon system access
  - Notifying the appropriate management-level users of the status of all in-process data, including ownership, process stage, etc.
  - Notifying the appropriate technical-level personnel the status of environmental monitoring sample records they 'own' upon system access, including in-process records, out-of-specification records, etc.

Data Processing (cont.)

- The system should manage the following aspects of an isolate identification system which works in conjunction with the environmental monitoring program
    - Allow for the recording of organism identification data for all pertinent processed sample locations
    - Upon detection of an out-of-specification condition regarding identified organism (objectionable) results, notify appropriate personnel of the aberrant event
    - Allow for user-defined macroscopic and microscopic organism characteristics
- The system should manage the completed environmental monitoring information from a review / approval standpoint, highlighted by the following;
    - Ensure that system users may not review / approve their own environmental monitoring data entries
    - Ensure that only system users with the proper access level can review / approve environmental monitoring data entries
    - Allow for authorized users to modify completed environmental monitoring records when warranted
    - Ensure that the record modification process requires 'change rationale' or equivalent from the authorized user
    - Provide the option to preclude the system from sending out-of-specification notifications for completed but not reviewed environmental monitoring data records if specified
    - Provide the system option to include completed but not reviewed environmental monitoring data records in reports if specified

Reporting

Report package

- The system should provide a reporting package for final results
  - The users should be allowed to select the report format, including, but not limited to:
    - Line graphs
    - Bar graphs
    - Histograms
  - The users should be allowed to select the query parameters, including, but not limited to:
    - Date range (sample, result, review)
    - Sampler
    - Building
    - Area
    - Location
    - Sample type
    - Result status (out-of-specification)
    - Worklist
    - Process
    - Media lot
    - Equipment ID
    - Product lot
- The system should provide a reporting package for identified organisms / isolates
  - The users should be allowed to select the report format, including, but not limited to:
    - Line graphs
    - Bar graphs
    - Histograms
  - The users should be allowed to select the query parameters, including, but not limited to:
    - Date range (sample, result, review)
    - Sampler
    - Building
    - Area
    - Location
    - Sample type
    - Organism
      - Gram stain
      - Out-of-specification organisms (objectionable)
      - Macroscopic results
    - Worklist
    - Process
    - Media lot
    - Equipment ID
    - Product lot

Reporting (cont.)

- The system should provide a reporting package for excursions
  - The users should be allowed to select the query parameters, including, but not limited to:
    - Date range (sample, result, review)
    - Sampler
    - Building
    - Area
    - Location
    - Sample type
    - Organism
    - Worklist
    - Process
    - Media lot
    - Equipment ID
    - Product lot
- The system should provide a reporting package for data result status (i.e. in-process, complete, reviewed)
  - The users should be allowed to select the query parameters, including, but not limited to:
    - Date range (sample, result, review)
    - Sampler
    - Building
    - Area
    - Location
    - Sample type
    - Organism
    - Worklist
    - Process
    - Media lot
    - Equipment ID
    - Product lot
    -

Data Transfer / Export

Data Transfer / Export

- The system should provide the capability to perform potential 'transfer / export / retrieval' of the system data to other data formats
    - The potential database formats the system should be 'compatible' with include:
        - Microsoft Excel
        - Microsoft Access
        - Oracle
        - SQL Server
        - Crystal Reports

21 CFR Part 11

Audit Trail

- The system should maintain an automatic (non-user modifiable) audit trail of all modifications made to the system. The audit trail should contain the following:
  - An entry identifying the type of modification made to the database / record, such as additions, deletions, modifications, review, etc.
  - An entry identifying the user performing the modification to the database / record
  - An entry noting the date / time of the database / record modification
  - The date / time included in the audit trail should be retrieved from the server system clock, not the local system clock
  - The entire database record being modified with the changed fields highlighted
  - The full manifestation of the name of the user performing the modification
  - The option to add a 'comment' to the audit trail record at the time of database record modification
- The audit records should be accessible only by authorized users
- The audit records should be able to be selected by date and category (component)
- The audit records should be reported in an 'easy-to-read' format (view and print)

Electronic Signatures

- The system should have the option to require an electronic signature prior to performing any modification to the database / records
- The electronic signature will require the entry of the username and password of the user performing the database modification
- The electronic signature will provide the option to enter information describing the 'meaning' of the signing
- The electronic signature will prevent the 'excising' (cut, copy, paste) of the electronic signature components between signings (i.e. a 'link' between the electronic signature and its database record)

EXHIBIT C

FUNCTIONAL SPECIFICATION

The system shall provide a method of importing / transferring data output from the Becton Dickinson Phoenix™ Automated Microbiology System into the Compliance Software Solutions Corp Environmental Monitoring Software System using a CFR Part 11 compliant data management environment. The BD Phoenix™ Automated Microbiology System is intended for the rapid identification (ID) and antimicrobial susceptibility testing (AST) of significant organisms.

1 SYSTEM REQUIREMENTS (EMSS)

The EMSS software is designed to reside on a local workstation and its database on a network server. The software can function without a network (i.e. with the database and application loaded on a local workstation).

The equipment recommendations are as follows:

Workstation:
- Pentium or any 32 bit microprocessor – minimum speed 400 MHz
- Microsoft Windows 98 Second Edition or higher
- 64 MB RAM (minimum) / 128 MB RAM recommended
- NIC (10/100 Ethernet)

Server:
- Minimum Pentium III
- 256 MB RAM (minimum) / 1.0 GB RAM recommended
- MSDE, SQL Server 7.0 or higher, or Oracle 8x or higher installed

2 USER REQUIREMENTS / FUNCTIONAL SPECIFICATIONS (EMSS/PHOENIX INTERFACE)

The interface should:

2.1 Ensure that the data transferred (uploaded) from the Phoenix™ to the EMSS retains its data integrity during the transfer process

2.2 Transfers (uploads) the data from the Phoenix™ using RS232 connectivity

2.3 Assume that the Phoenix™ input Accession ID is the same string value as the EMSS Sample ID

2.4 Provide for the creation of 'supplemental' Sample ID labels for placing on the Phoenix™ test panels (to facilitate entry of Accession ID by Phoenix™ user)

2.5 Transfers (uploads) the Accession ID, Observation ID, Organism ID, Test ID, Sequence ID, Instrument ID, Test Start date, Test Complete date, Data Sent date, and Comments information from the Phoenix™

2.6 The Phoenix™ data will be transferred (uploaded) to the EMSS database automatically according to the settings on the Phoenix™ Communications Configuration screen (i.e. Solicited, Send on Finalization, Send on Completion, Send as Available, Send at Fixed Time)

2.7 The Accession ID, Observation ID, Organism ID, Instrument ID, and Test Complete date are non-modifiable and automatically populated within the EMSS interface (Routine Organism ID form)

2.8 The Source field within the organism data is automatically populated with Phoenix™ if the organism data in question is coming from the Phoenix™

2.9 The organism data is subject to the same CFR Part 11 / user security level constraints including audit trail, change functions, etc. as the environmental monitoring data within EMSS

2.10 The Phoenix™ data will be accessible in reports, etc. in the same manner as EMSS environmental data

2.11 Provide for data / error management in the event that communication is lost between the Phoenix™ and the EMSS database (i.e. transfer/upload queue/buffer, EMSS query of the 3100-record Phoenix™ database, etc.)

2.12 Provide for data / error management in the event that the transferred (uploaded) Organism ID code does not exist in the 'organism library'

REQUIREMENTS DEFINITION

The system shall provide a method of importing / transferring data output from the Becton Dickinson Phoenix™ Automated Microbiology System into the Compliance Software Solutions Corp Environmental Monitoring Software System using a CFR Part 11 compliant data management environment. The BD Phoenix™ Automated Microbiology System is intended for the rapid identification (ID) and antimicrobial susceptibility testing (AST) of significant organisms.

1 SYSTEM REQUIREMENTS (EMSS)

The EMSS software is designed to reside on a local workstation and its database on a network server. The software can function without a network (i.e. with the database and application loaded on a local workstation).

The equipment recommendations are as follows:

Workstation:
- Pentium or any 32 bit microprocessor – minimum speed 400 MHz
- Microsoft Windows 98 Second Edition or higher
- 64 MB RAM (minimum) / 128 MB RAM recommended
- NIC (10/100 Ethernet)

Server:
- Minimum Pentium III
- 256 MB RAM (minimum) / 1.0 GB RAM recommended
- MSDE, SQL Server 7.0 or higher, or Oracle 8x or higher installed

2 USER REQUIREMENTS (EMSS/PHOENIX INTERFACE)

The interface should:

- Ensure that the data transferred (uploaded) from the Phoenix™ to the EMSS retains its data integrity during the transfer process
- Transfers (uploads) the data from the Phoenix™ using RS232 connectivity
- Assume that the Phoenix™ input Accession ID is the same string value as the EMSS Sample ID
- Provide for the creation of 'supplemental' Sample ID labels for placing on the Phoenix™ test panels (to facilitate entry of Accession ID by Phoenix™ user)
- Transfers (uploads) the Accession ID, Observation ID, and Organism ID information only from the Phoenix™
- The Phoenix™ data will be transferred (uploaded) to the EMSS database automatically according to the settings on the Phoenix™ Communications Configuration screen (i.e. Solicited, Send on Finalization, Send on Completion, Send as Available, Send at Fixed Time)
- The Accession ID, Observation ID, and Organism ID are non-modifiable and automatically populated within the EMSS interface (Routine Organism ID form)
- The Source field within the organism data is automatically populated with Phoenix™ if the organism data in question is coming from the Phoenix™
- The organism data is subject to the same CFR Part 11 / user security level constraints including audit trail, change functions, etc. as the environmental monitoring data within EMSS
- The Phoenix™ data will be accessible in reports, etc. in the same manner as EMSS environmental data
- Provide for data / error management in the event that communication is lost between the Phoenix™ and the EMSS database (i.e. transfer/upload queue/buffer, EMSS query of the 3100-record Phoenix™ database, etc.)
- Provide for data / error management in the event that the transferred (uploaded) Organism ID code does not exist in the 'organism library'

Configuration Management Plan

The EMSS Phoenix Interface project will be managed according to the roles stated in item 2.3 in SOP QA-402. The members of the project team are Tony Castellano, Tim Joyce, Victoria Galliani, Alla Arens, Cindy Jones, and George Levinson.

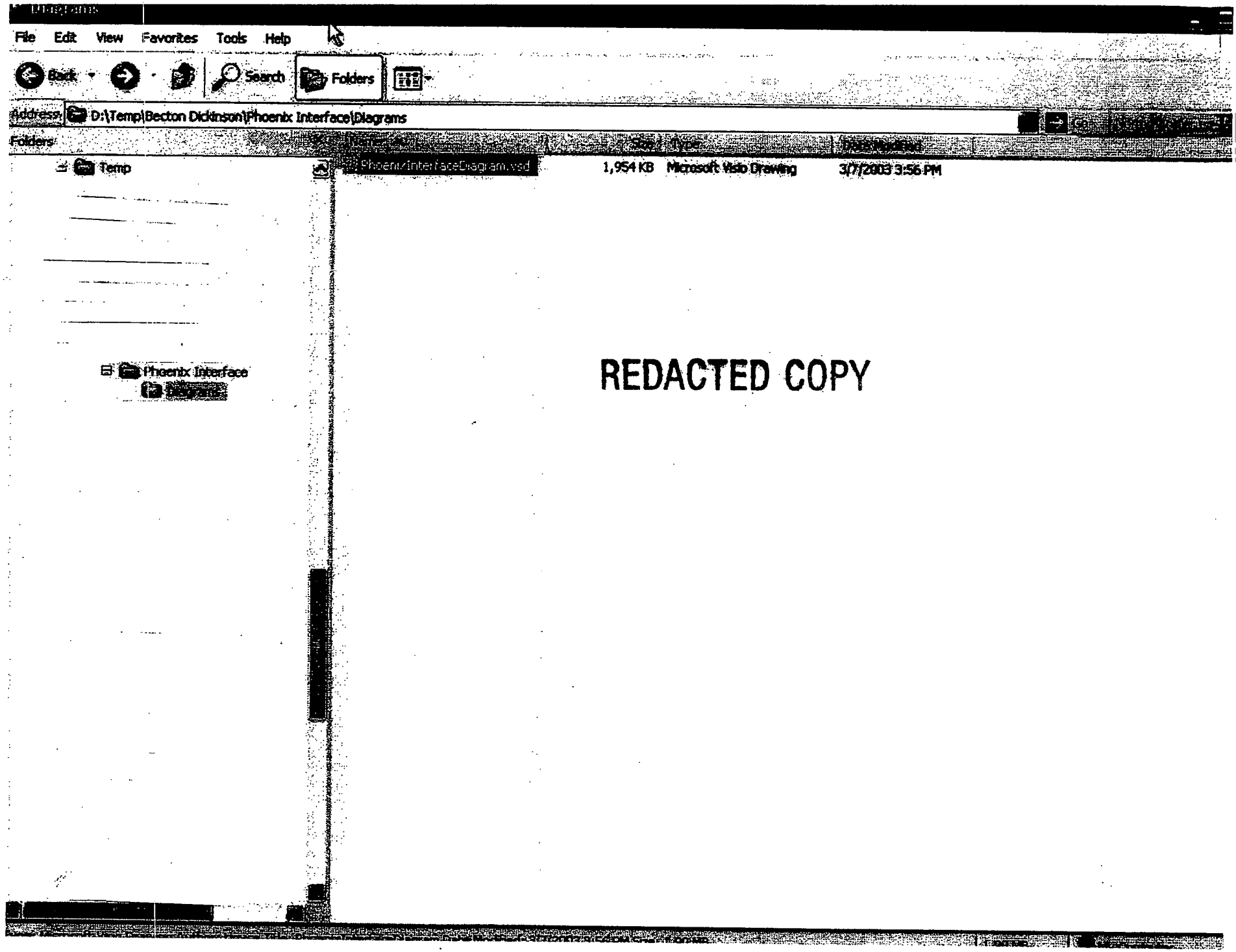
The following table lists the roles identified for distribution of project tasks and the team members assigned to them:

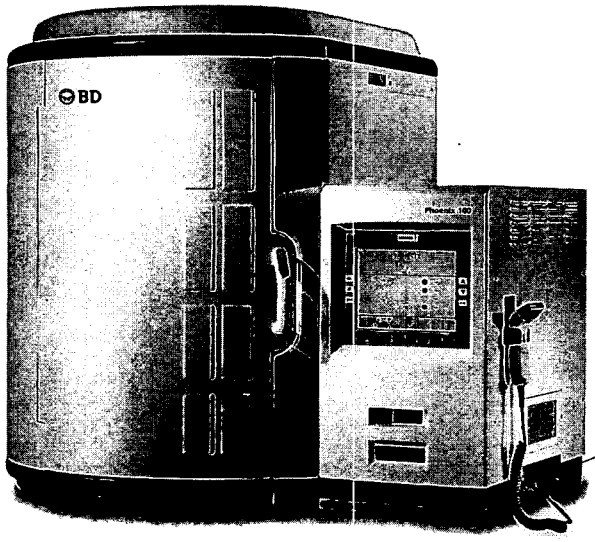
PROJECT ROLE DISTRIBUTION

Roles	Project Team Members
Marketing	George Levinson*
Program Management	Tim Joyce*
Information Systems	Tony Castellano*
Quality Assurance	Victoria Galliani*
Validation	Alla Arens, Tony Castellano, Victoria Galliani*, Cindy Jones, Tim Joyce*, George Levinson, Todd Daniels, Rob Beard
Installation and Technical Support	Tony Castellano*, Tim Joyce*

* Denotes primary member(s) of role

EXHIBIT D


File Edit View Favorites Tools Help
Back
Search
Folders
Address D:\Temp\Becton Dickinson\Phoenix Interface\Diagrams
Folders
Temp
Phoenix Interface
Diagrams
PhoenixInterfaceDiagram.vsd
1,954 KB
Microsoft Visio Drawing
3/7/2003 3:56 PM
REDACTED COPY



RS232

Serial / Ethernet converter

10 Base T

Workstation (with / without EMSS)

10 Base T

Ethernet

10 Base T

10 Base T

EMSS Oracle / SQL Database Server

EMSS Workstation

The system as we envision it will work as follows:

When a panel is ready to be placed into the Phoenix instrument, the operator will scan the sequence number as usual, and will also scan in the accession number which will be printed on a second label generated by the EMSS. We need to 'match up' the Phoenix panel (identified by the sequence number) with the corresponding EMSS database record (identified by the 'sampleID') and the user-entered Phoenix accession number meets the requirement perfectly (format, size, etc.). When the panel processing is complete and the panel record is sent (independent of the setting in the Phoenix LIS Config screen) to the RS232 port, the Serial / Ethernet converter (our proposed 'black box') will write the record to the EMSS database through a standard networked workstation. The EMSS then will retrieve the organism identification data from the transmitted Phoenix panel record for the appropriate sampleID. This data can then be reported along with other EMSS data and is also 'protected' via the EMSS' built-in audit trail and electronic signature compliance.

We will be building in 'protection' for the data transmission process if the communications are lost between the Phoenix and the ethernet, along with the condition that the identified organism is not defined in the EMSS Species database table.

EXHIBIT E

Millennium Electronics, Inc.

Innovations In Technology

NETWORK SERIAL PORT ARRAY

DESIGN PACKAGE

FOR

COMPLIANCE SOFTWARE

SOLUTIONS CORPORATION

750 BUNKER COURT, SUITE 200
VERNON HILLS, IL 60061

JUNE 12, 2003

Millennium Electronics, Inc.

Innovations In Technology

Table of Contents

Millennium Electronics, Inc.

Innovations In Technology

NETWORK SERIAL PORT ARRAY

BILL OF MATERIALS

MILLENNIUM ELECTRONICS INC.

Prepared by Andrew K. Miraldi 6/12/2003

BOM FOR ASM3035-REV-1 MULTI-PORT MOTHER BOARD.

ITEM	PART No.	DESC.	VENDOR P/N	MFR	REF.DES.	QTY PER
1	A3035-REV.1	P.W.B.-MULTI-PORT MOTHER BOARD	A3035-REV.1	MULTI-CIRCUITS	PWB	1
2	101-0520	IC-RCM3200 RABBITCORE MODULE	101-0520	RABBIT SEMI.	U3	1
3	MAX3223EAP	IC-3,3V RS232 TRANCEIVER SOIC-20	MAX3223EAP	MAXIM	U1,2	2
4	LM2575S-ADJ	IC-SWITCHING REGULATOR ADJ T0-263	LM2575S-ADJ	NATIONAL	VR1	1
5	DF02S	IC-1A-200V SMT BRIDGE RECTIFIER	DF02SGITR-ND	GENERAL SEMI.	BR1	1
6	SK33-7	DIODE-SCHOTTKY-30V-3A-SMC	SK33DITR-ND	DIODES INC.	D1	1
7	SMAJ12CA	TVS-12V-400W-BI-DIR	SMAJ12CADICT-ND	DIODES INC.	TVS1	1
8	ECJ-2VB1E104K	CAP-CERAMIC CHIP 0.1uF-25V-XR7-0805	PCC1828CT-ND	PANASONIC	C1-11,14,15	13
9	AVS35V100	CAP- ALUM-SMT 100uF-35V-"F" SIZE	AVS35V100	CDE	C12	1
10	AVS16V330	CAP- ALUM-SMT 330uF-16V-"F" SIZE	AVS16V330	CDE	C13	1
11	ECS-T1DX685R	CAP- TANT-SMT 6.8uF-20V-3528-"B" SIZE	P11323CT-ND	PANASONIC	C16	1
12		RES CHIP 3.30K-1%-0805		ANY	R1	1
13		RES CHIP 2.00K-1%-0805		ANY	R2	1
14		RES CHIP 330-5%-0805		ANY	R4-11	8
15		RES CHIP 10K-5%-0805		ANY	R3,12,13	8
16	SMI-0330-S	IND-330uH-SMT	SMI-0330-S	PREM	L1	1
17	3425L150DR	POLY-FUSE-1.5A-SMT-3425L	3425L150DR	LITTLEFUSE	U11-8	1
18	EVQ-PF003M	SWITCH MOMENTARY PB R/A	P10877S-ND	PANASONIC	SW1	1
19	PJ-102A	POWER JACK-2.11 1MM PWB PINS	CP-102A-ND	CUI-STACK	J1	1
20	10-89-1101	HEADER 0.1" 2x5 POS. MALE	WM6810-ND	MOLEX	J6,7	2
21	747844-4	CONNECTOR-DB-9 R/A FEMALE	A203301-ND	AMP	J2-5	4
22	PS2M43-217GB	HEADER 2mm 2x17 POS. FEMALE	416-0008-WEB	RABBIT SEMI.	for U3	2
23	551-0209	LED T-1 R/A PWB MNT GRN	350-1084-ND	DIALIGHT	LED1,LED6	2
24	551-0309	LED T-1 R/A PWB MNT YEL	350-1086-ND	DIALIGHT	LED2-5,LED7	5
25	551-0409	LED T-1 R/A PWB MNT RED	350-1088-ND	DIALIGHT	LED8	1

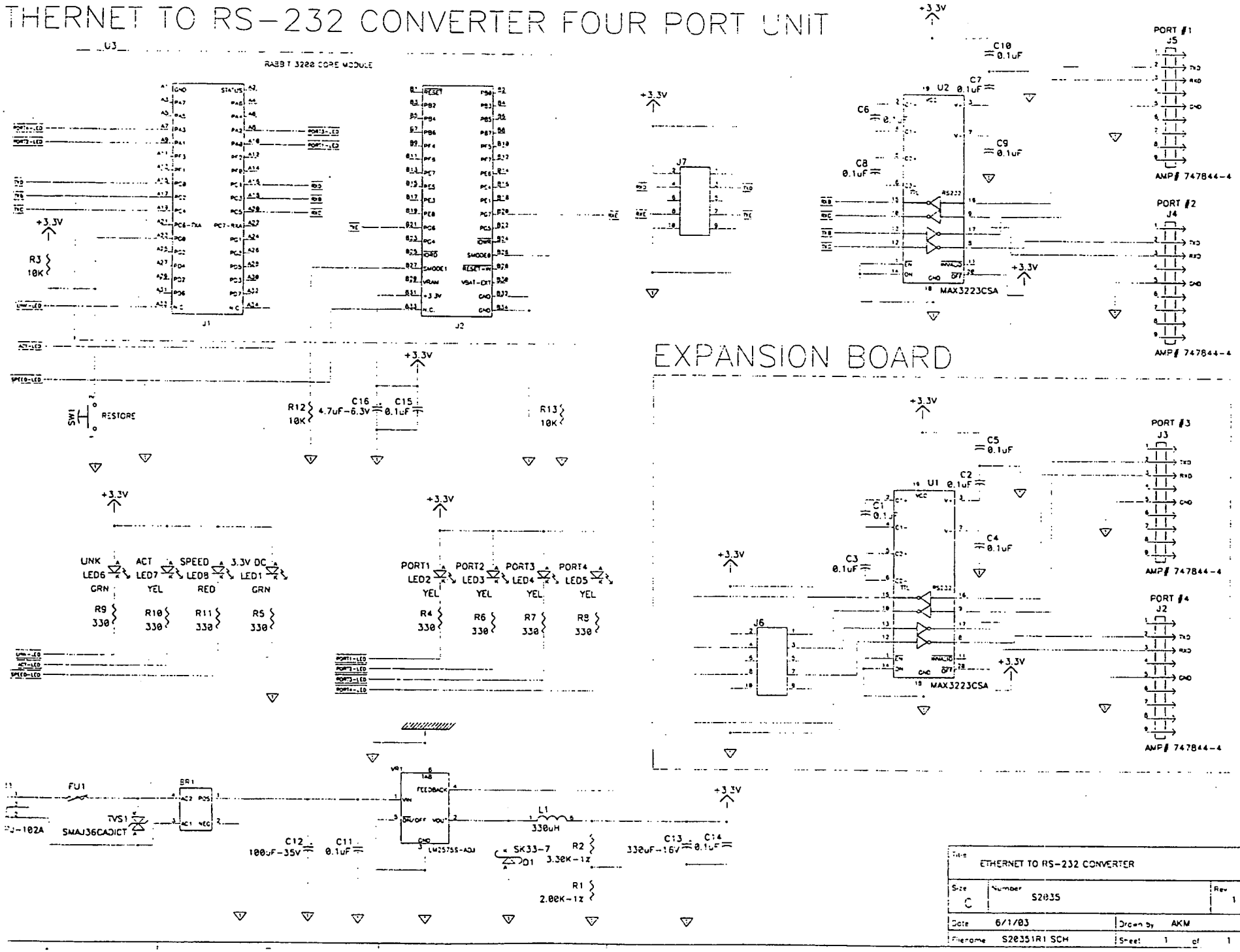
Millennium Electronics, Inc.

Innovations In Technology

NETWORK SERIAL PORT ARRAY

SCHEMATIC

THERNET TO RS-232 CONVERTER FOUR PORT UNIT



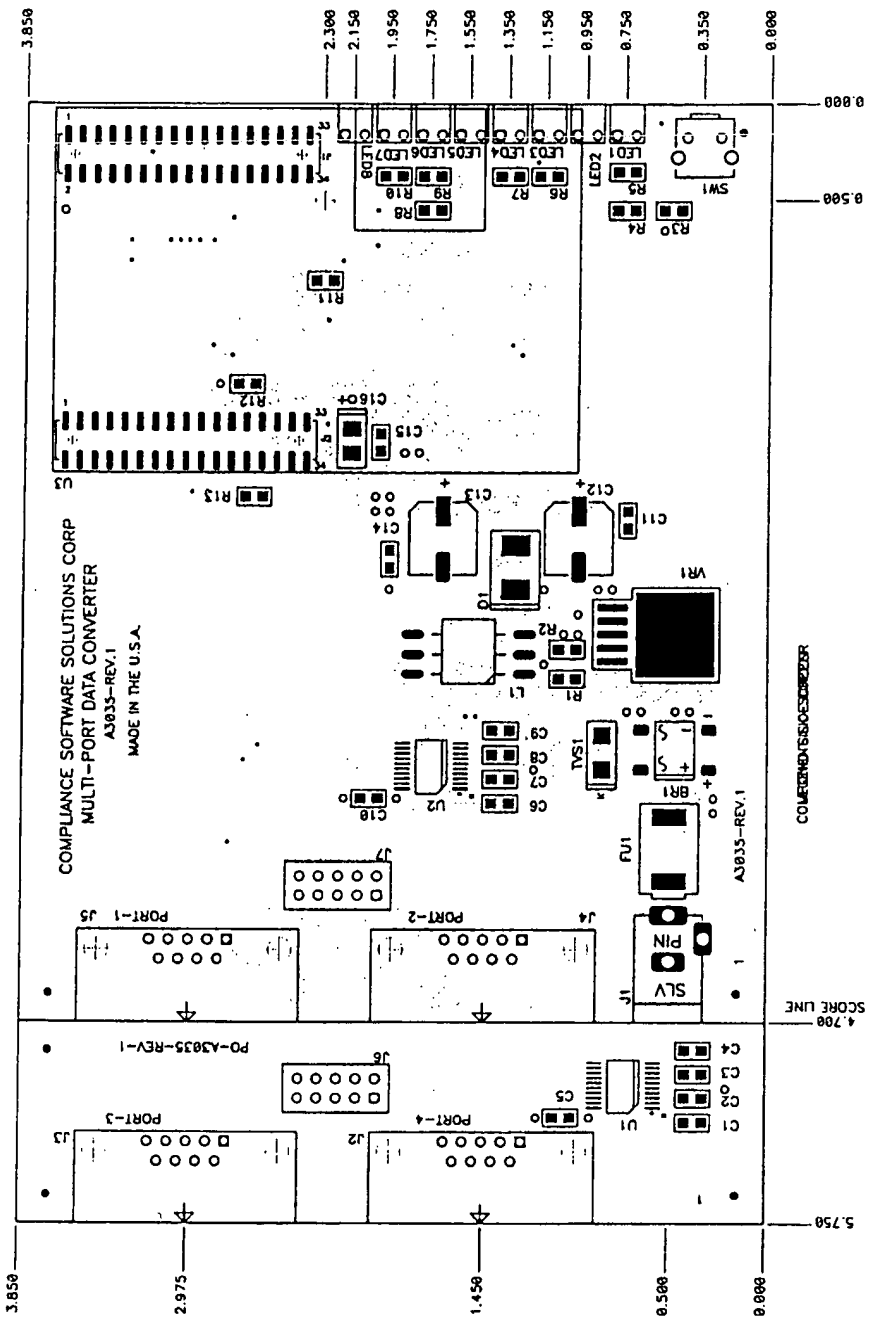
Title	ETHERNET TO RS-232 CONVERTER		
Size	C	Number S2035	Rev 1
Date	6/1/03	Drawn by AKM	
Filename	S2035R1.SCH	Sheet 1 of 1	

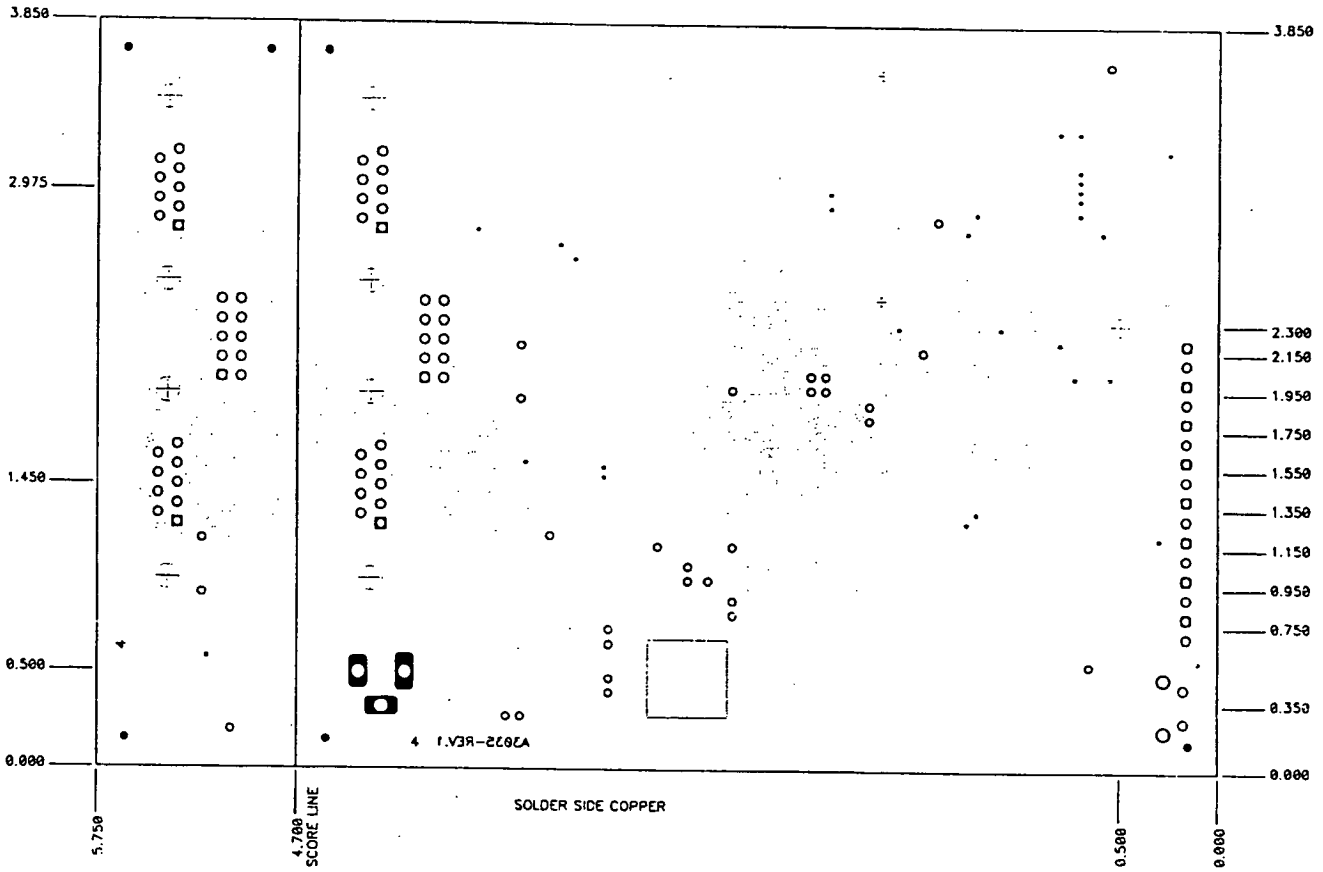
Millennium Electronics, Inc.

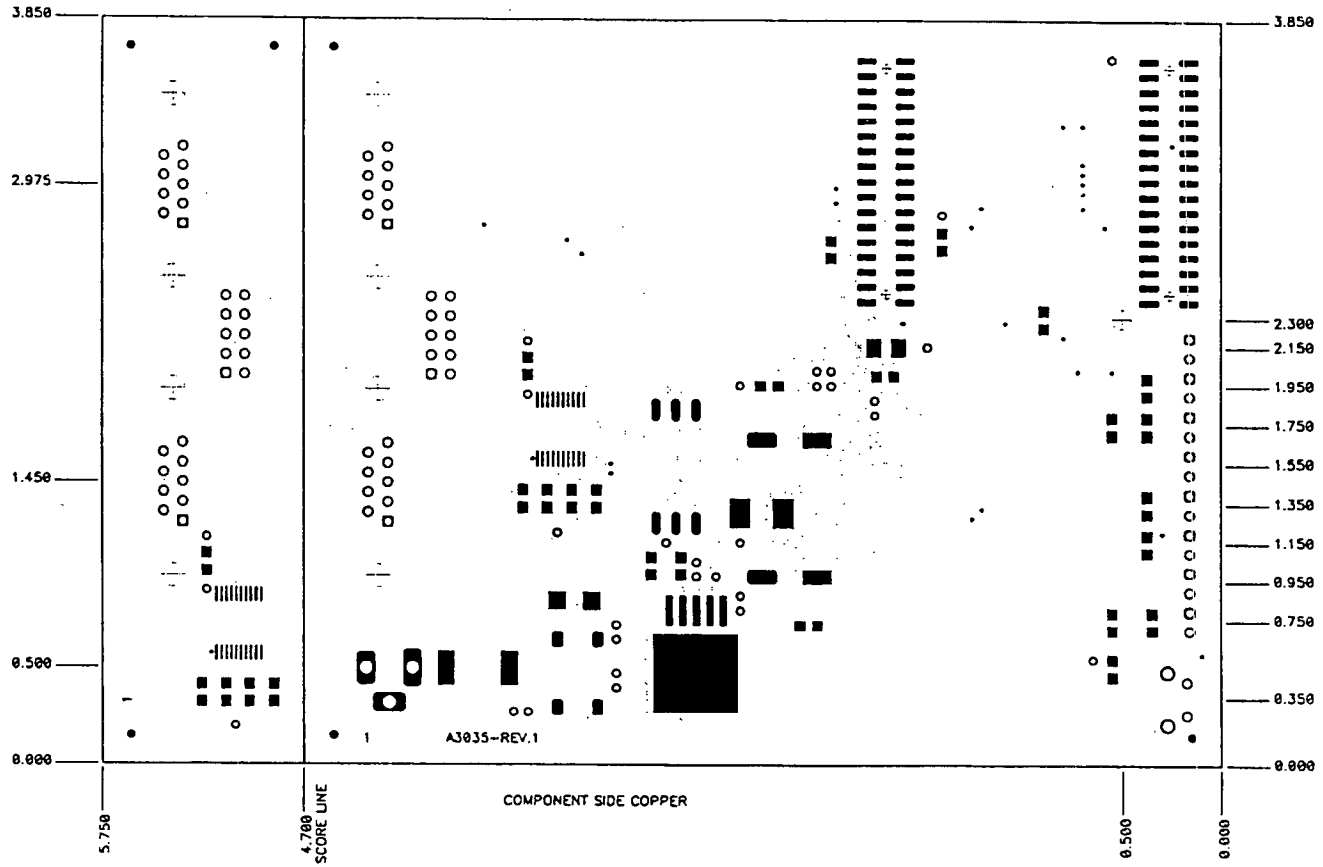
Innovations In Technology

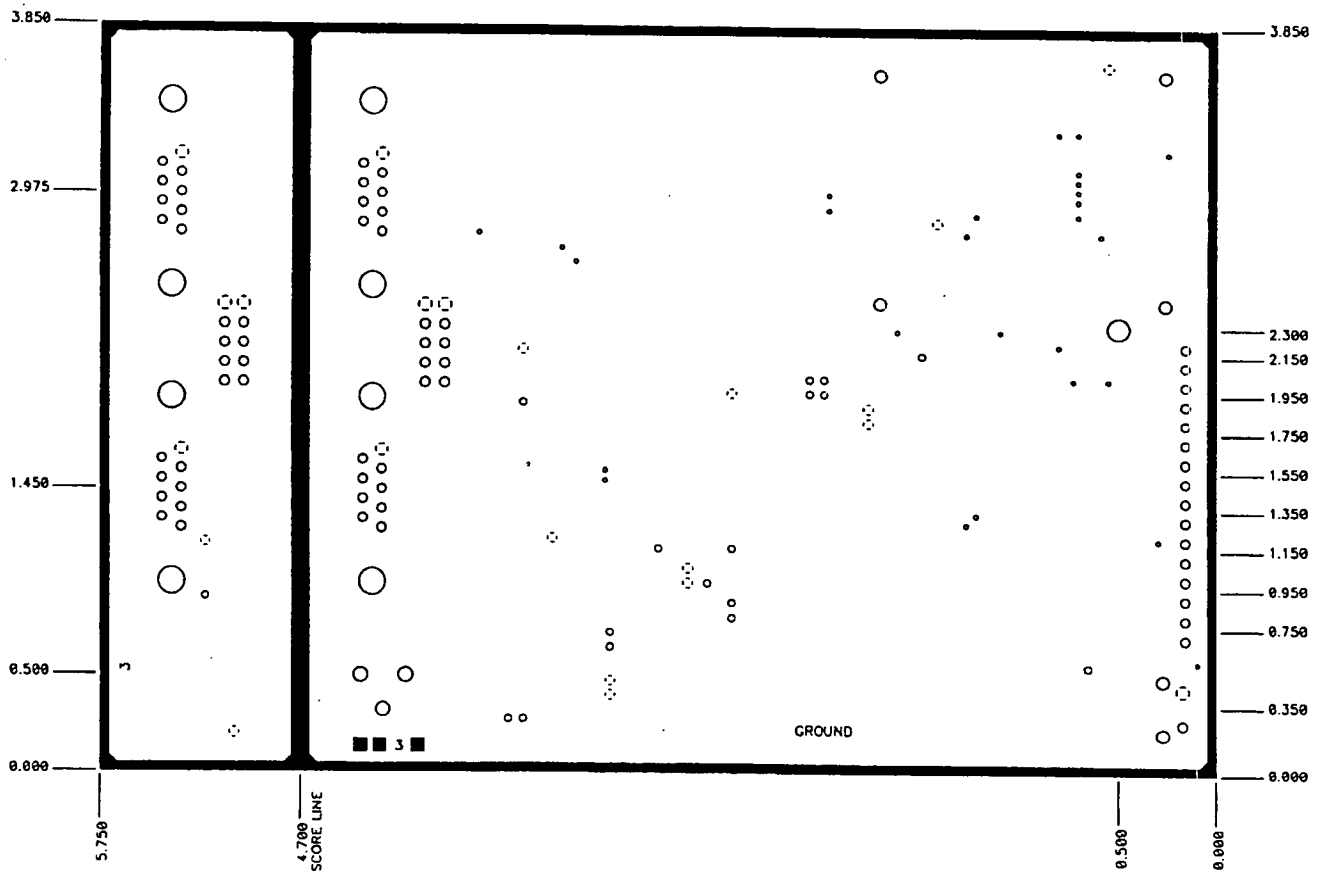
NETWORK SERIAL PORT ARRAY

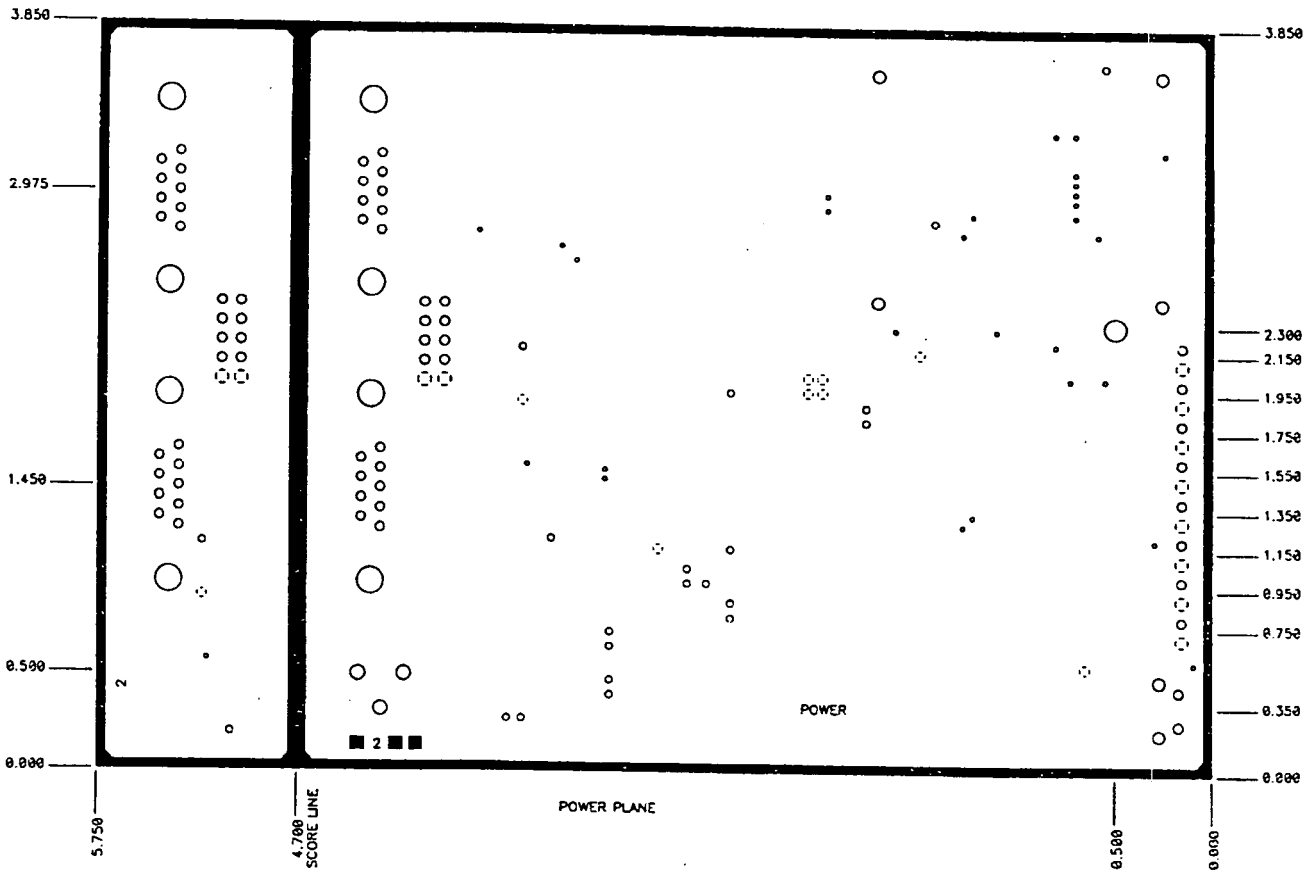
PRINTED WIRING BOARD ARTWORK

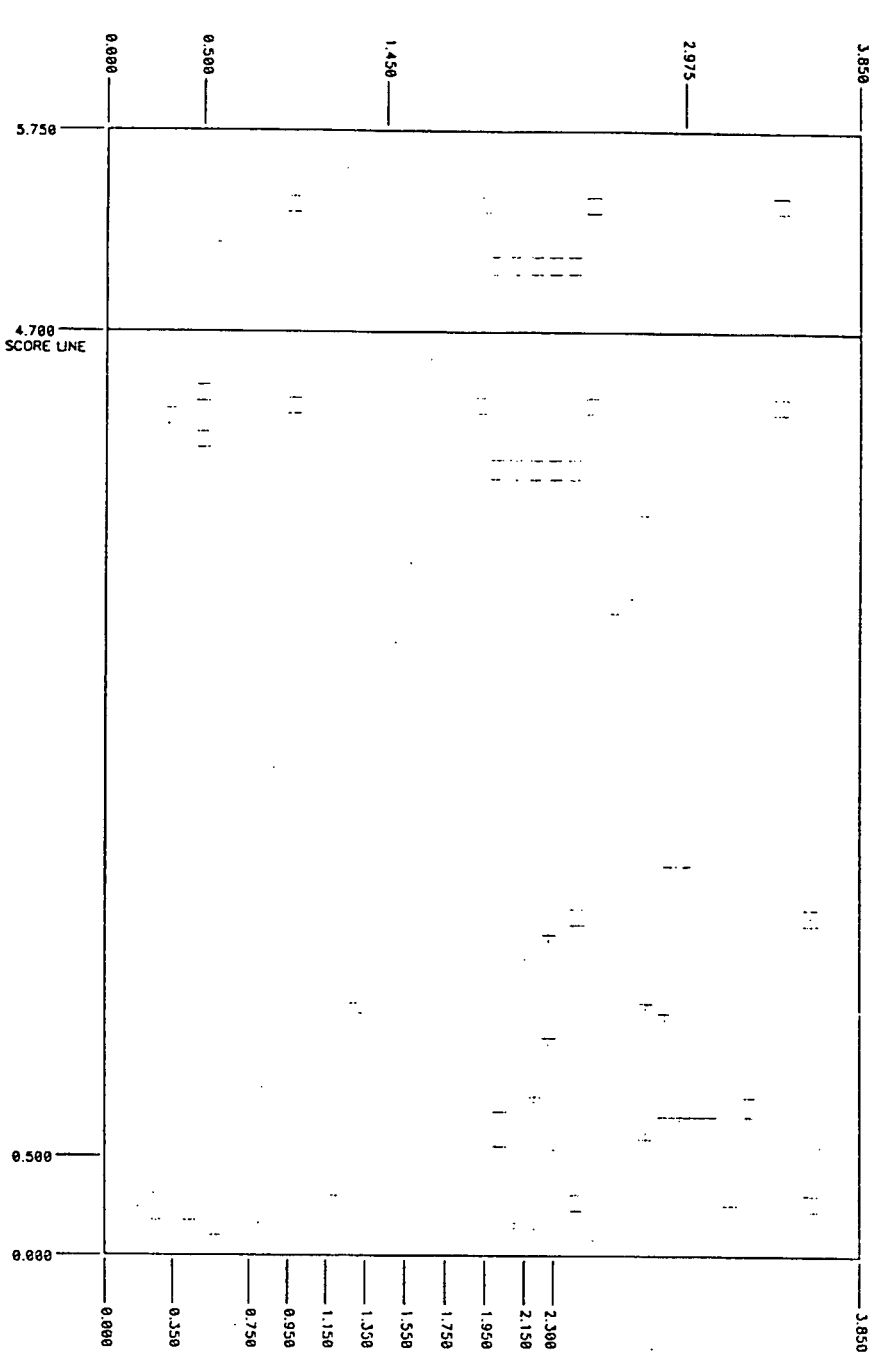

COMPLIANCE SOFTWARE SOLUTIONS CORP
MULTI-PORT DATA CONVERTER
A3035-REV.1
MADE IN THE U.S.A.
PO-A3035-REV-1
A3035-REV.1
PORT-1
PORT-2
PORT-3
PORT-4
SLV
PIN
SCORE LINE
3.850
2.300
2.150
1.950
1.750
1.550
1.350
1.150
0.950
0.750
0.350
0.000
0.000
0.500
4.700
5.750
3.850
2.975
1.450
0.500
0.000


3.850
2.975
1.450
0.500
0.000
5.750
4.700
SCORE LINE
SOLDER SIDE COPPER
0.500
0.000
3.850
2.300
2.150
1.950
1.750
1.550
1.350
1.150
0.950
0.750
0.350
0.000


3.850
2.975
1.450
0.500
0.000
1
1
A3035-REV.1
2.300
2.150
1.950
1.750
1.550
1.350
1.150
0.950
0.750
0.350
0.000
5.750
4.700
SCORE LINE
0.500
0.000
COMPONENT SIDE COPPER


3.850
2.975
1.450
0.500
0.000
3.850
2.300
2.150
1.950
1.750
1.550
1.350
1.150
0.950
0.750
0.350
0.000
3
GROUND
3
5.750
4.700
SCORE LINE
0.500
0.000


3.850
2.975
1.450
0.500
0.000
5.750
4.700
SCORE LINE
0.500
0.000
2
POWER
2
POWER PLANE
3.850
2.300
2.150
1.950
1.750
1.550
1.350
1.150
0.950
0.750
0.350
0.000



Drill Table			
(inch)	Symbol	Quantity	Plated
2	+	29	Yes
8	X	30	Yes
6	Y	52	Yes
9	T	22	Yes
2	⋈	2	Yes
2	⧖	4	No
2	⊡	3	Yes
0	◇	1	No
5	⊟	8	No

R ORDER
MPONENT SIDE COPPER
WER PLANE
OUND PLANE
LDER SIDE COPPER

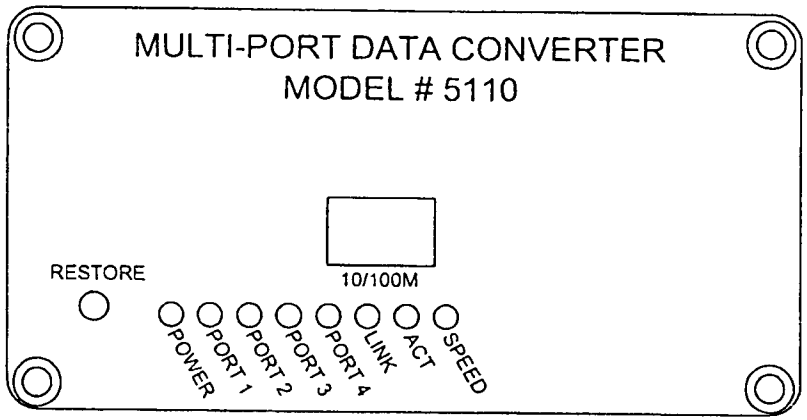
Millennium Electronics, Inc.

Innovations In Technology

NETWORK SERIAL PORT ARRAY

MECHANICAL DRAWINGS (FRONT)

FRONT M3037-REV-0

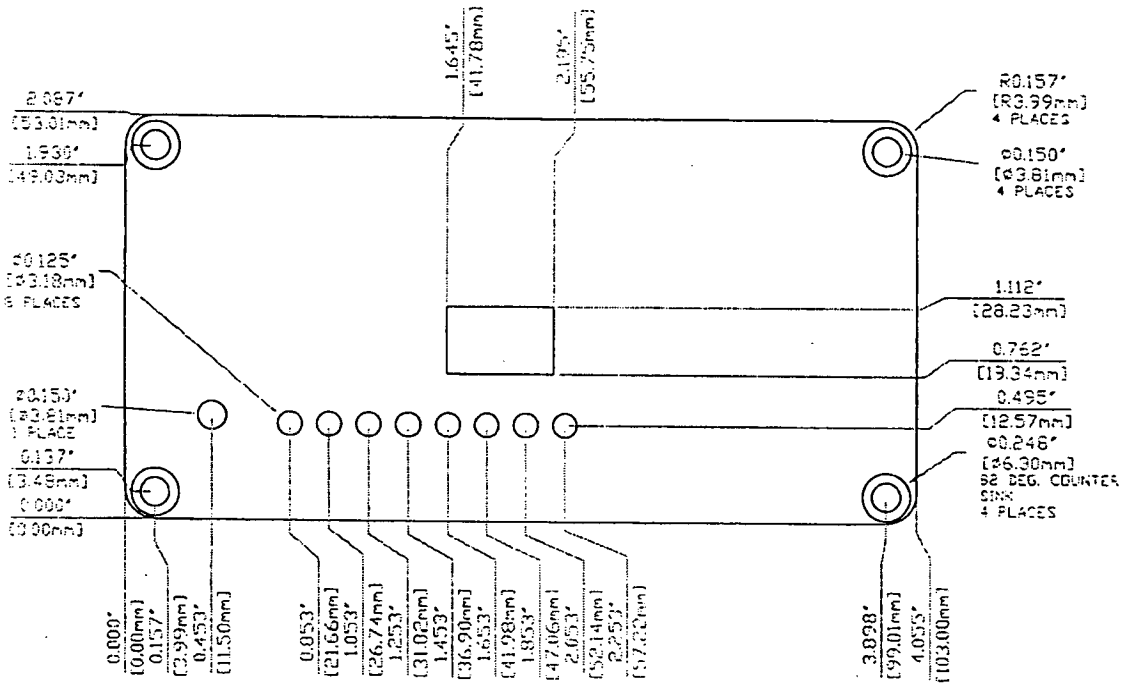


NOTES

1. MATERIAL TO BE 0.060" ALUMINUM.
2. FINISH TO BE PAINTED PANTONE#xxxxx.
3. SILKSCREEN TO Mxxxx-REV-0.
4. BREAK ALL SHARP CORNERS AND EDGES.

TOLERENCE	REV.	DATE	DISCRIPTION OF CHANGE/ECO	BY	CHK'D	MILLENNIUM ELECTRONICS
DECIMAL ± .010	0	02/05/03	1st RELEASE	AKM		TITLE: MULIT-PORT FRONT PANEL
FRACTAL ± .020						CUST: COMPLIANCE SOFTWARE SOLUTIONS CORP.
ANGULAR ± .10%						FILE: M3037IR0.DWG DATE: 05/02/03
						DRAWN BY: AKM SCALE: 1:1 CHK'D:
						MTRL: SEE NOTES M3037
						SHEET 1 OF 1

FRONT M3037-REV-0

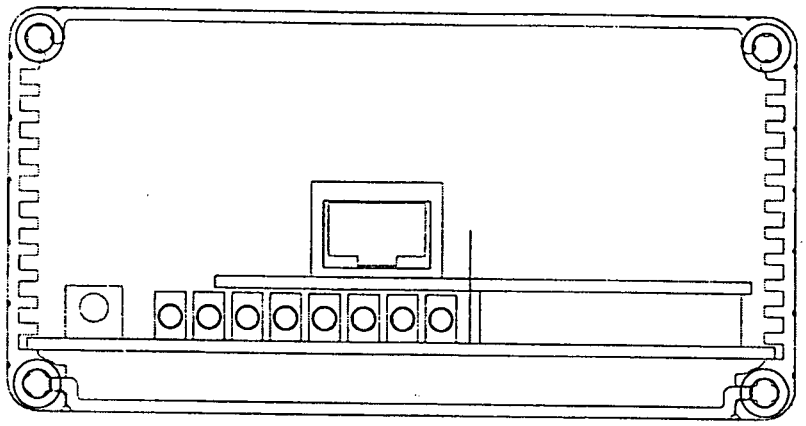


NOTES

1. MATERIAL TO BE 0.060" ALUMINUM.
2. FINISH TO BE PAINTED PANTONE#xxxxx.
3. SILKSCREEN TO Mxxxx-REV-0.
4. BREAK ALL SHARP CORNERS AND EDGES.

TOLERENCE	REV	DATE	DISCRIPTION OF CHANGE/ECO	BY	CHK'D	MILLENNIUM ELECTRONICS
DECIMAL ± .010	0	02/05/03	1st RELEASE	AKM		TITLE: MULIT-PORT FRONT PANEL
FRACTAL ± .020						CUST: COMPLIANCE SOFTWARE SOLUTIONS CORP.
ANGULAR ± .10°						FILE: M3037R0.DWG DATE: 05/02/03
						DRAWN BY: AKM SCALE: 1:1 CHK'D:
						MTRL: SEE NOTES M3037 0
						SHEET 1 OF 1

FRONT M3037-REV-0



NOTES

1. MATERIAL TO BE 0.060" ALUMINUM.
2. FINISH TO BE PAINTED PANTONE#xxxxx.
3. SILKSCREEN TO Mxxxx-REV-0.
4. BREAK ALL SHARP CORNERS AND EDGES.

TOLERENCE	REV.	DATE	DISCRIPTION OF CHANGE/ECO	BY	CHK'D
DECIMAL ± .010	0	02/05/03	1st RELEASE	AKM	
FRACTAL ± .020					
ANGULAR ± .10%					

MILLENNIUM ELECTRONICS

TITLE: MULIT-PORT FRONT PANEL		
CUST: COMPLIANCE SOFTWARE SOLUTIONS CORP.		
FILE: M3037iR0.DWG	DATE: 05/02/03	
DRAWN BY : AKM	SCALE:N	CHK'D:
MTRL: SEE NOTES	M3037	0
SHEET 1 OF 1		

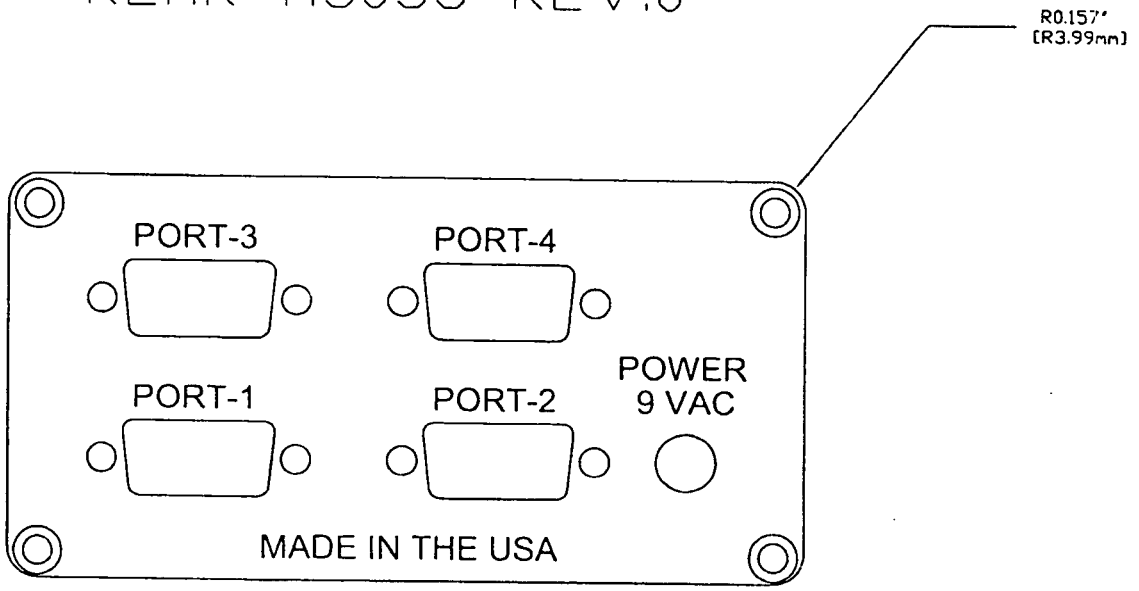
Millennium Electronics, Inc.

Innovations In Technology

NETWORK SERIAL PORT ARRAY

MECHANICAL DRAWINGS (REAR)

REAR M3038-REV.0

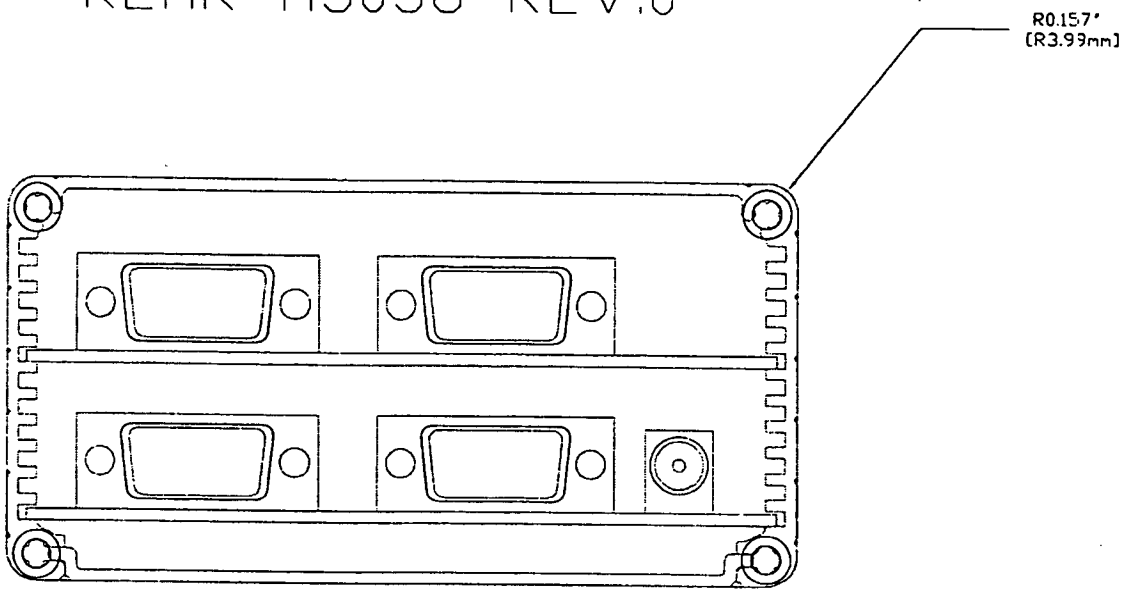


NOTES

1. MATERIAL TO BE 0.060" ALUMINUM.
2. FINISH TO BE PAINTED PANTONE#xxxxx.
3. SILKSCREEN TO Mxxxx-REV-0.
4. BREAK ALL SHARP CORNERS AND EDGES.

TOLERENCE	REV.	DATE	DISCRIPTION OF CHANGE/ECO	BY	CHK'D	MILLENNIUM ELECTRONICS		
DECIMAL ± .010	0	02/05/03	1st RELEASE	AKM		TITLE: MULTI-PORT REAR PANEL		
FRACTAL ± .020						CUST: COMPLIANCE SOFTWARE SOLUTIONS CORP.		
ANGULAR ± .10°						FILE: M3036IRQ.DWG	DATE: 5/2/03	
						DRAWN BY: AKM	SCALE: 1:1	CHK'D:
						MTRL: SEE NOTES	M3038	0
						SHEET 1 OF 1		

REAR M3038-REV.0

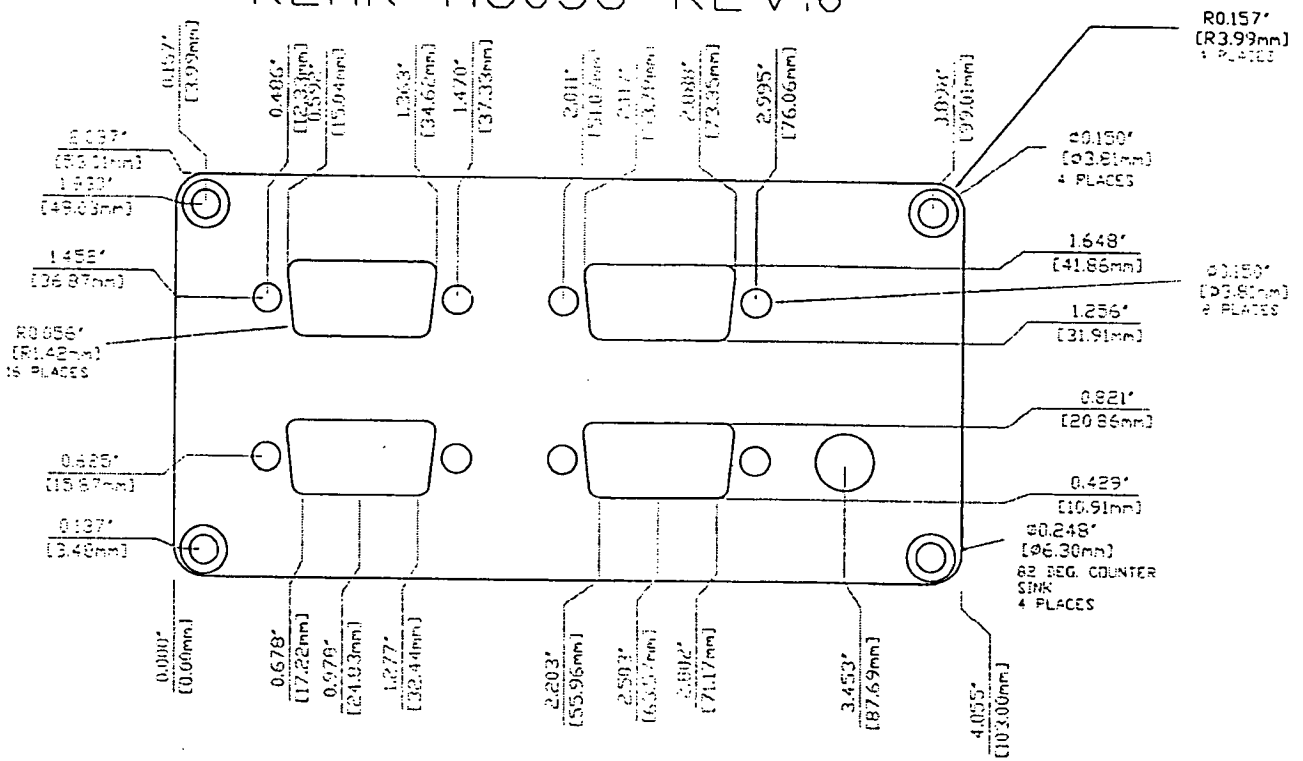


NOTES

1. MATERIAL TO BE 0.060" ALUMINUM.
2. FINISH TO BE PAINTED PANTONE#xxxxx.
3. SILKSCREEN TO Mxxxx-REV-0.
4. BREAK ALL SHARP CORNERS AND EDGES.

TOLERENCE	REV	DATE	DISCRIPTION OF CHANGE/ECO	BY	CHK'D	MILLENNIUM ELECTRONICS
DECIMAL ± .010	0	02/05/03	1st RELEASE	AKM		TITLE: MULTI-PORT REAR PANEL
FRACTAL ± .020						CUST: COMPLIANCE SOFTWARE SOLUTIONS CORP.
ANGULAR ± .10°						FILE: M3038IR0.DWG DATE: 5/2/03
						DRAWN BY: AKM SCALE: 1:1 CHK'D:
						MTRL: SEE NOTES M3038 0
						SHEET 1 OF 1

REAR M3038-REV.0



NOTES

1. MATERIAL TO BE 0.060" ALUMINUM.
2. FINISH TO BE PAINTED PANTONE#xxxxx.
3. SILKSCREEN TO Mxxxx-REV-0.
4. BREAK ALL SHARP CORNERS AND EDGES.

TOLERENCE	REV	DATE	DISCRIPTION OF CHANGE/ECO	BY	CHK'D	MILLENNIUM ELECTRONICS
DECIMAL ± .010	0	02/05/03	1st RELEASE	AKM		TITLE: MULTI-PORT REAR PANEL
FRACTAL ± .020						CUST: COMPLIANCE SOFTWARE SOLUTIONS CORP.
ANGULAR ± .10°						FILE: M3038R0.DWG DATE: 5/2/03
						DRAWN BY: AKM SCALE: 1:1 CHK'D:
						MTRL: SEE NOTES M3038
						SHEET 1 OF 1 0

NETWORK SERIAL PORT ARRAY

SOFTWARE INTERFACE SPECIFICATION

Updated
APRIL 9, 2003

Table of Contents

Section 1. - Overview

1.1. Target Application

1.1.1. The network serial port array includes software targeted for use in any Windows operating system (XP, 2K, NT, Me, 98, 95)

1.1.2. The software library encapsulates the network protocol required to interface to the 4-port network serial devices in a system and give the user application simple read/write and setup functions for interacting with the serial ports.

1.1.3. The software takes the form of a Microsoft "ActiveX control" which is an object that can easily be embedded in a custom Visual Basic application or other forms based application. This includes popular HMI software such as LabView and WonderWare. As an extreme example, it could even be embedded in a Microsoft Excel spreadsheet running macros.

1.1.4. ActiveX controls "reveal" their available commands (methods) and properties to the design environment for the application being built. For example, the VB Toolbox can be modified to include the control, which can then be dropped in to any form.

1.1.5. For Visual C++ programmers and other developers who prefer to use DLLs instead of ActiveX controls, a DLL is also available with identical commands as the ActiveX control. This documentation is intended to describe the ActiveX control.

1.2. Also Included

1.2.1. Sample VB Program: A sample Visual Basic .NET program that simulates a multi-channel text terminal is supplied that shows how to use the ActiveX control commands. Complete source code is included with step-by-step instructions for how part of it was created.

1.2.2. A Windows utility that allows the user to detect and configure the Network IP addresses of these serial port arrays.

1.3. Modes of Operation

1.3.1. The target application must first call setup commands to configure each port that will be used. A port is referenced by the IP address of its controller and the port number (0-3) within the 4-port array. Each port can be configured differently.

1.3.2. Configuration options for the ports include Baud-rate, parity start/stop bits etc. But they also include special timing and buffering parameters that effect performance of serial-over Ethernet communications. The application developer may select to use the default parameters and ignore this extra complexity but they are available for fine-tuning of performance.

1.3.3. Sending data is always done in Non-blocking mode. This means that the function call to send the data returns immediately so that the application is not slowed down even though the actual output of the serial bytes will not be completed until several milliseconds later (seconds, if the amount is large)

1.3.4. Receiving data is also done in non-blocking mode. Incoming data can be polled by the outer application or the ActiveX control can be set up to trigger an "event handler" only when data arrives.

Section 2. - Commands

2.1. Setup

2.1.1. A separate control instance must be added to the application for each separate serial port. The port is configured by setting its properties and then by calling the enable method to open the serial port.

2.1.2. Property: **BaudRate** as Integer

- Sets the baud rate for the port.
- Type Integer, valid values are 2400, 4800, 9600, 14400, 38400.
- Default Value = 9600

2.1.3. Property **IpAddress** as String

- Indicates the IP address of the 4-port serial controller that owns this port
- Valid values are any network IP address in dot notation like '192.168.1.14'
- Default Value = None (Most methods will fail if this is not configured)

2.1.4. Property **PhysicalPort** as Integer

- Indicates the physical port number within the 4-port serial controller.
- Default Value = 0, Valid Values 0-3

2.1.5. Property **Parity** as integer

- Sets Parity handling for the port
- 0= No Parity, 1=Odd, 2=Even, 3=Mark, 4=Space
- Default value = 0, No Parity

2.1.6. Property **StopBits** as integer

- Sets number of stop bits for the port
- 1=1Bit, 2=2 bits, 3=1.5 bits

2.1.7. Property **RtsControl** as integer

- Sets the RTS (Request to send line) behavior. Note, this usually has no effect on simple RS232 communications.
- 0=Always off, 1= Always On, 2 = On only when transmitting (useful for RS485 converters)
- Default = 2

2.1.8. Method **Enable() As Integer**

- This enables the serial port with a configuration defined by the properties mentioned above. All Send and Receive commands will fail when the port is not enabled successfully. If another NetSerial control in the application (or in another application running on the same network) has already enabled this same physical port on the controller with this same Ip Address, then this call will fail. If IpAddress has not been set prior to calling enable, then the call will fail. If any other setup property has not been set prior to calling enable, then its default value will be used.

- Return Value: 1 if successful, 0 if failed. Call GetErrorCode or GetErrorMsg to see detailed information about the failure.

2.1.9. Method **Disable()**

- This disables the port. All pending send and receive operations are canceled. The configuration properties can be changed while the port is disabled and will take effect when re-enabled.

2.2. Sending

2.2.1. Sending over the serial ports is always non-blocking. As soon as any of the commands below are issued, the data is buffered and the function returns immediately. The buffer is then sent out over the serial port.

2.2.2. It is difficult to have precise control over the bytes in a buffer to be sent in Visual basic String handling tools are fooled when a NULL byte is inserted, it is often stripped out. Some string handling routines add modify CR\LF character pairs. If this is not a problem, then the SendString function can be used to send a string's contents over the serial port. If it is a problem, then the SetBuffer and SendBuffer commands can be used to send precisely constructed messages of various binary values

2.2.3. Method **SendString (theString as String) AS Integer**

- This accepts the string contents and adds them to the control's transmit buffer to be sent. The function returns immediately without waiting for the transmission to actually be completed. If the previous message has not had enough time to be fully sent, then the current message is put on the end. If the current message must go out immediately without waiting for prior messages, then first call the PurgeTxBuffer method.
- Return Value: 1 if successful, 0 if failed. Call GetErrorCode or GetErrorMsg to see detailed information about the failure.

2.2.4. Method **SetBuffer (Index AS Integer, theByte AS Byte)**

- Several of these calls can be used to prepare a buffer to be sent using SendBuffer. Index is a zero-based offset from the start of the buffer and theByte is placed at that index. If an index is skipped in the buffer, then its contents will be whatever was contained there prior to the last SendBuffer command. On startup, the buffer contents are initially all zeros.

2.2.5. Method **SendBuffer (Length AS Integer) AS Integer**

- This sends the first 'Length' bytes of the buffer that was built up using SetBuffer commands. The buffer used is a special separate buffer just for this command, it should not be confused with the internal transmit buffer. This command behaves just like SendString in how it adds the contents to be sent to any existing messages still in the transmit buffer.
- Return Value: 1 if successful, 0 if failed. Call GetErrorCode or GetErrorMsg to see detailed information about the failure.
- Example: The following set of function calls will send the 4-byte sequence 255, 63, 0, 255 over the serial port.
  - SetBuffer (0, 255)
  - SetBuffer (1,63)
  - SetBuffer (2,0)
  - SetBuffer (3, 255)

2.3. Receiving

2.3.1. Receiving from the serial port is also always non-blocking. ReceiveString, ReceiveBuffer, or ReceiveByte can be called any time to get whatever incoming bytes have been received in the internal receive buffer. These calls return immediately. If nothing has been received then they return zero bytes. CountAvailable returns the number of bytes that are available to be retrieved.

2.3.2. Applications can simply poll incoming bytes by continuously calling ReceiveString or ReceiveBuffer. But the control also has the ability to work asynchronously. Applications can take advantage of the OnReceiveEvent function, which is called whenever a predetermined number of bytes have come in or when a timeout period has expired. The behavior of OnReceiveEvent depends on the parameters used in a call to RequestReceive or AlwaysReceive

2.3.3. Method **ReceiveString (MaxLength AS Integer) As String**

- This command takes whatever has recently been received in the receive buffer up to MaxLength and returns it as a string. If there are more than MaxLength bytes waiting in the receive buffer, then these will remain in the buffer after the function call and can be retrieved later with another ReceiveString call.
- If no bytes are available then an Empty string is returned.

2.3.4. Method **ReceiveBuffer (MaxLength AS Integer) AS Integer**

- This command takes whatever has recently been received in the receive buffer up to MaxLength and copies it to a special buffer where it can be analyzed using calls to GetBufferAt. This can be useful for parsing messages that contain bytes, which are not easily manipulated as strings in VB.
- Returns the number of bytes actually copied and available for use by GetBufferAt commands. If nothing has been received lately, then this function returns zero.

2.3.5. Method **ReceiveByte() AS Integer**

- This returns the next byte in the receive buffer. This can be called repeatedly to read an incoming message one byte at a time. When the receive buffer is empty, the integer result is –1 (similar to end-of-file).
- Returns –1 if there are no more bytes to read, otherwise returns the byte value 0-255.

2.3.6. Method **GetBufferAt(Index AS Integer) AS Integer**

- This is used after a call to ReceiveBuffer to examine the buffers contents. Index is a zero-based offset from the start of the buffer to be returned. Care should be taken not to specify an index that is past the number of bytes actually copied in the call to ReceiveBuffer because this can return an undefined value.
- Returns –1 if the specified index is past the end of what was actually buffered, otherwise returns the byte value 0-255.

2.3.7. Method **CountAvailable() AS Integer**

- This returns the number of bytes waiting in the receive buffer that can be read using ReceiveBuffer, ReceiveString, or repeated calls to ReceiveByte.

2.3.8. Method **PurgeRxBuffer() As Integer**

- This clears any unread contents out of the receive buffer. Thus the next call to ReceiveString or ReceiveBuffer will return no new bytes.
- Returns the number of unread bytes that were in the buffer before it was cleared.

2.4. Asynchronous Receiving

2.4.1. Method **RequestReceive(Length AS Integer, Timeout AS Integer)**

- This function tells the control to begin receiving the number of bytes requested by the Length parameter and to trigger the application's OnReceiveEvent handler function when they have come in. If the specified Timeout value (in milliseconds) elapses before the requested number of bytes is received, then the OnReceiveEvent handler is triggered as well.
- Within OnReceiveEvent, the application can look at the RequestStatus property to determine if the request ended because of a timeout or because all of the requested bytes were received. The application can also call CountAvailable to determine how many bytes were actually received. If this is less than the number requested, then a timeout can be assumed.
- If the Timeout value is zero, this is interpreted as an Infinite timeout. Thus if no bytes come in, OnReceiveEvent will never be triggered.
- RequestReceive is considered a one-shot function. OnReceiveEvent will only be triggered once for each call to RequestReceive. If more bytes come in afterwards, OnReceiveEvent will not be triggered again (as they are in AlwaysReceive – see below) If RequestReceive is called again before the last call was completed, the last request is simply cancelled.

2.4.2. Property **RequestStatus** AS Integer

- This property can be read to determine the status of the last RequestReceive command.
- 0 = Request is not yet complete. The timeout has not expired and the requested number of bytes has not yet been received.
- 1 = Requested number of bytes have been received and can be read using ReceiveString etc.
- 2 = Timeout expired before the requested number of bytes could be received.
- 3 = Communication error with the controller – possibly loss of a network connection or no power to the controller.

2.4.3. Method **AlwaysReceive (MaxBlockLength AS Integer, GapTime AS Integer)**

- This tells the control to begin continuously receiving bytes and continuously notifying the application of new arrivals by triggering OnReceiveEvent (see above)
- If the bytes come in and then the receive line goes silent for longer than GapTime (milliseconds), then this is considered a separate incoming message and OnReceiveEvent is triggered.
- If bytes are streaming in without a gap, then every time MaxBlockLength bytes have been received, OnReceiveEvent is called.
- Within OnReceiveEvent, CountAvailable can be called to see how much has actually come in. The RequestStatus property will always be 0 unless there is a network problem in which case the value will be 3.

2.4.4. Method **StopAlwaysReceive( )**

- This cancels the AlwaysReceive mode which may be in progress and stops this from triggering OnReceiveEvent until AlwaysReceive or RequestReceive is called again

2.4.5. Event **OnReceiveEvent( )**

- As described in various cases above, this event is triggered in the end application in response to AlwaysReceive and RequestReceive commands when requested number of bytes is received.

Section 3. - Application Development

3.1. Adding to a VB.NET application

3.1.1. Run the quick setup program to register the control on the computer being used to develop the application. VB will handle redistribution automatically. This will not need to be done on the end-user's computer

3.1.2. Under the Tools menu, select "Customize Toolbox"

3.1.3. Under the "Com Components" screen, locate the "NetSerial" control. Click its checkbox, and click OK.

3.1.4. The NetSerial control should now be available in the Windows Forms pane in the Forms toolbox for the visual basic project.

3.1.5. Add multiple NetSerial controls to one of the Forms in the project. – One for each port that will be used.

3.1.6. Resize these controls to be very small. The control is intended to be non-visible, but if it is left visible, it will display a small activity indicator icon when serial communications are taking place. Select the control and change its Visible Property to suit.

3.1.7. View Code for the form and on the select any of the NetSerial components that were just added, and then select OnReceiveEvent from the declarations drop list. This adds a serial event handler function to the form which will be triggered when non-blocking read/write events are completed for that port

3.1.8. Function calls can be made to any of the NetSerial1 member functions for setup, reading, and writing.

- SendBuffer(4)

3.1.9. Method **PurgeTxBuffer() As Integer**

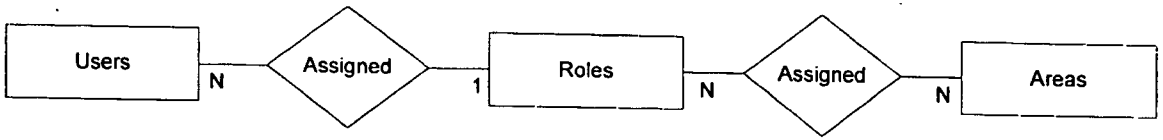
- This clears any unsent contents out of the transmit buffer. Thus the next call to SendString or SendBuffer will begin sending its contents out immediately.
- Returns the number of unsent bytes that were in the buffer before it was cleared.

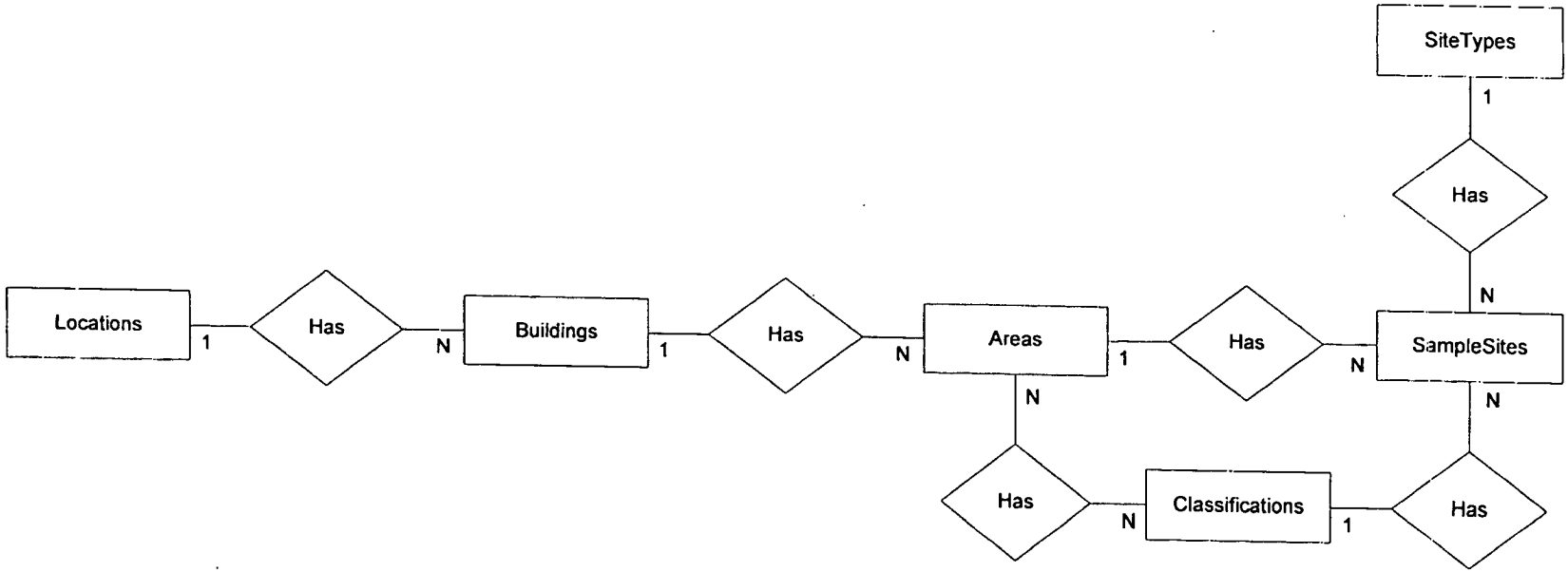
EXHIBIT F

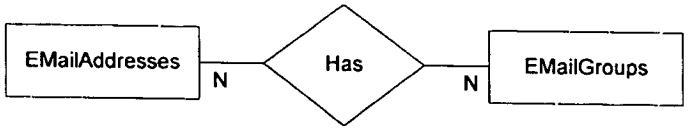
Product: EMS v. 3.0		Document: Entity Relationship Diagrams
Created By: Alla Arens		Revised By: Tony Castellano
Version: 10	Date: 5/09/2003	Attachment A

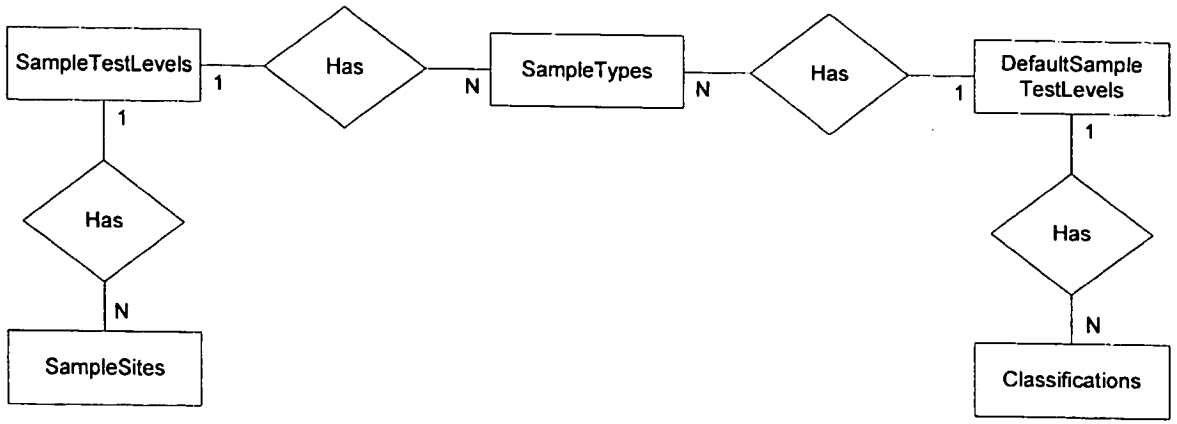
Revision Summary

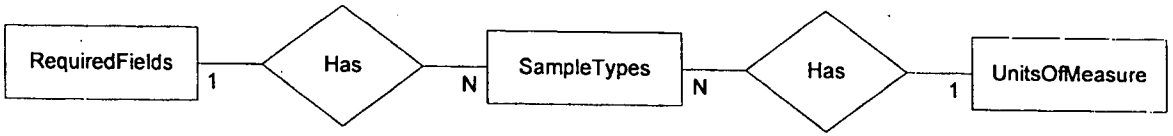
- Added the PhoenixLog entity.
- Added the PhoenixOrganisms entity.

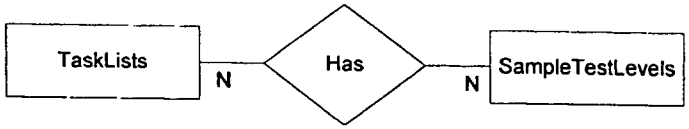

Users
N
Assigned
1
Roles
N
Assigned
N
Areas


SiteTypes
1
Has
N
Locations
1
Has
N
Buildings
1
Has
N
Areas
1
Has
N
SampleSites
N
N
Has
N
Classifications
1
Has

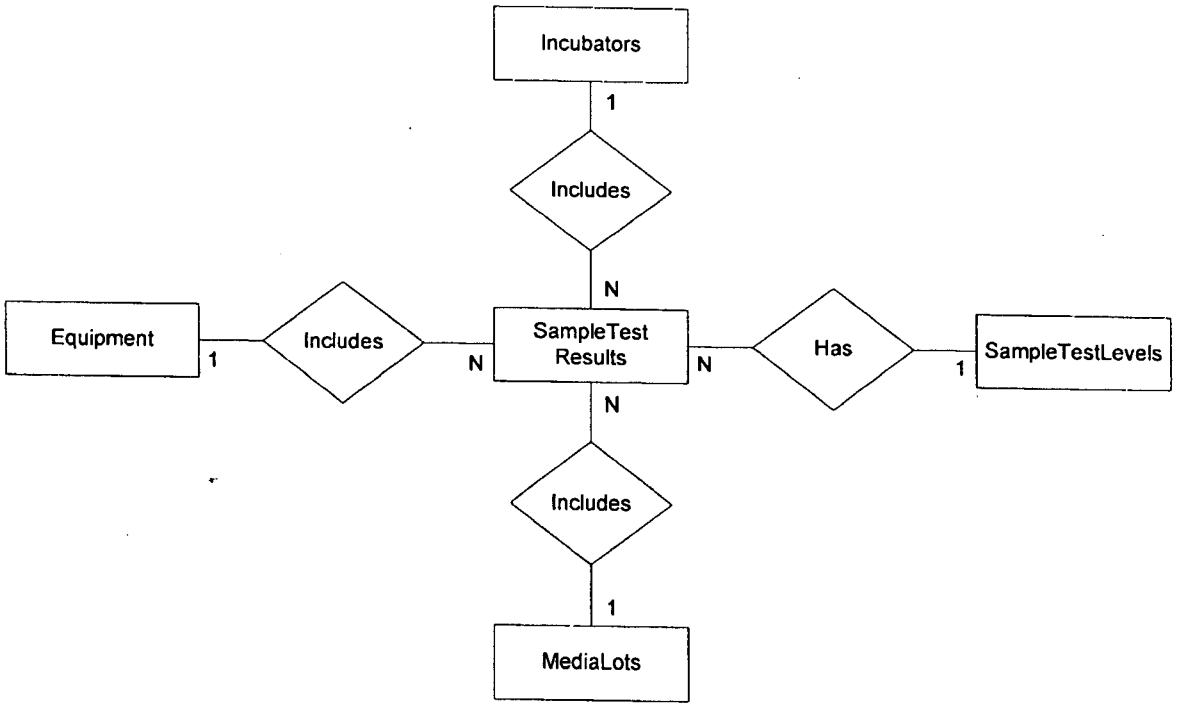

EMailAddresses
N
Has
N
EMailGroups

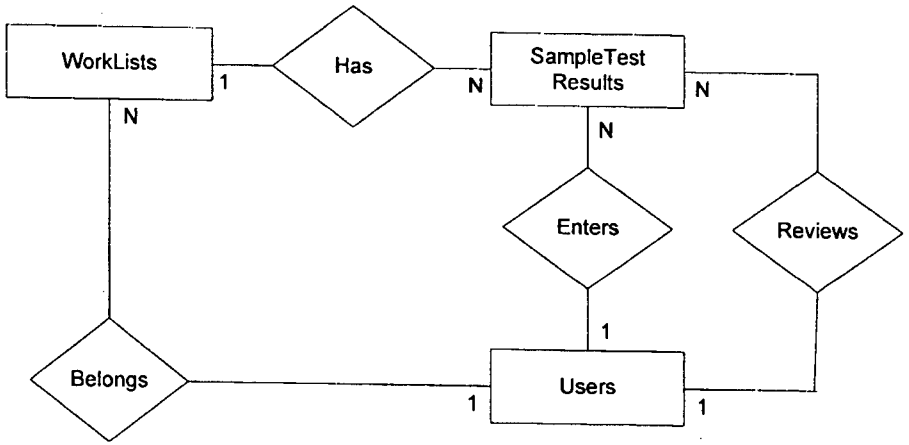

SampleTestLevels
1
Has
N
SampleTypes
N
Has
1
DefaultSample TestLevels
1
Has
N
Classifications
1
Has
N
SampleSites

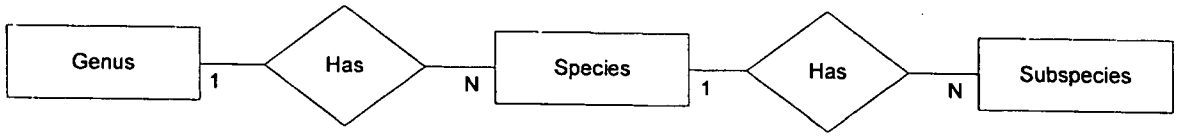

RequiredFields
1
Has
N
SampleTypes
N
Has
1
UnitsOfMeasure

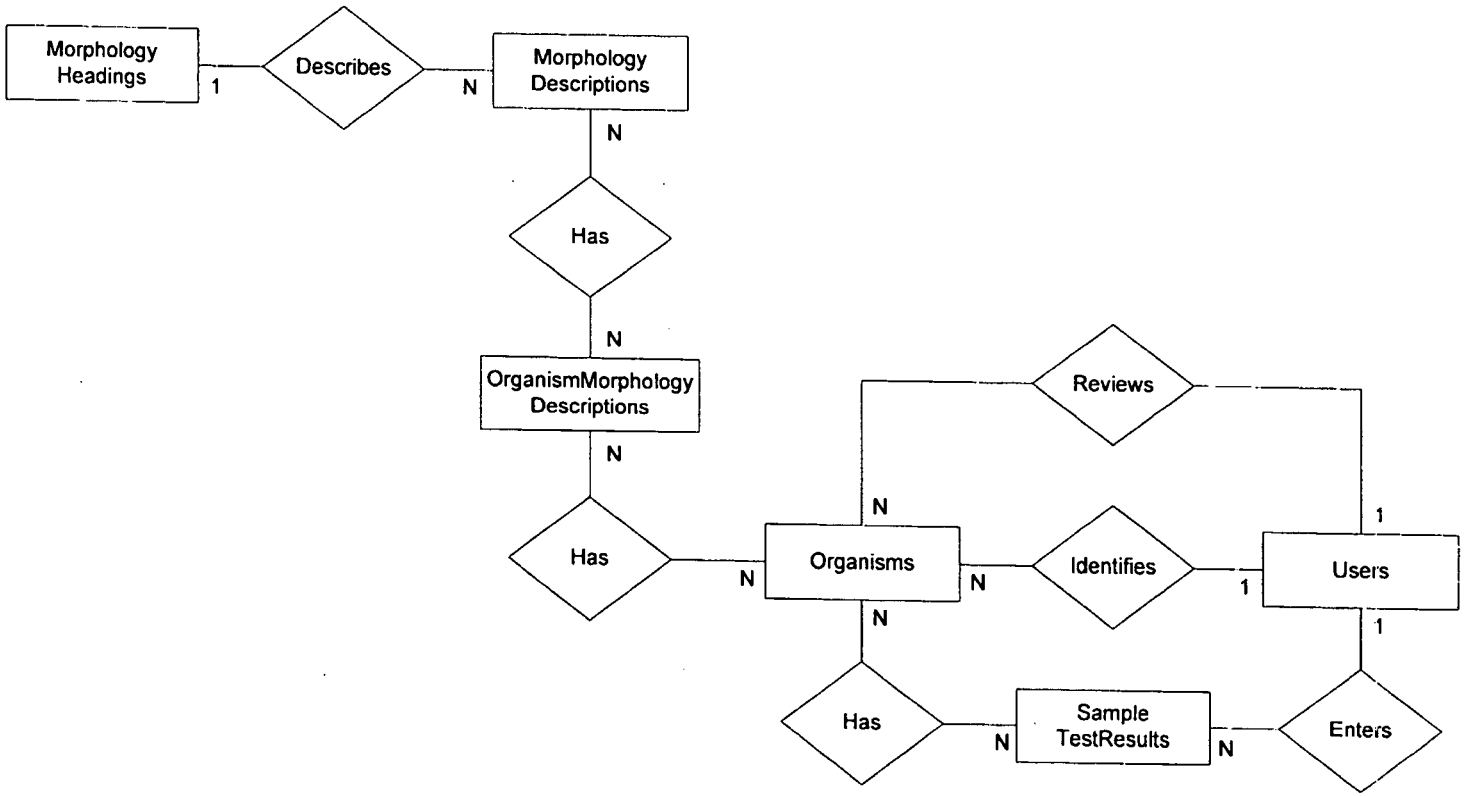

TaskLists
N
Has
N
SampleTestLevels

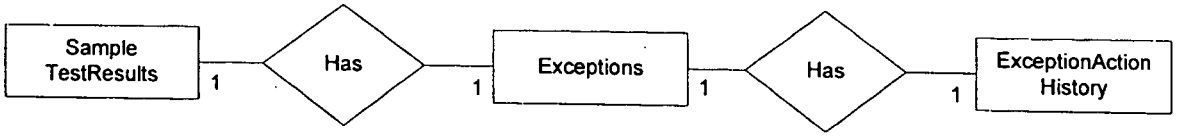

Users
1
Belongs to
N
WorkLists
N
Belongs to
1
EmailGroups

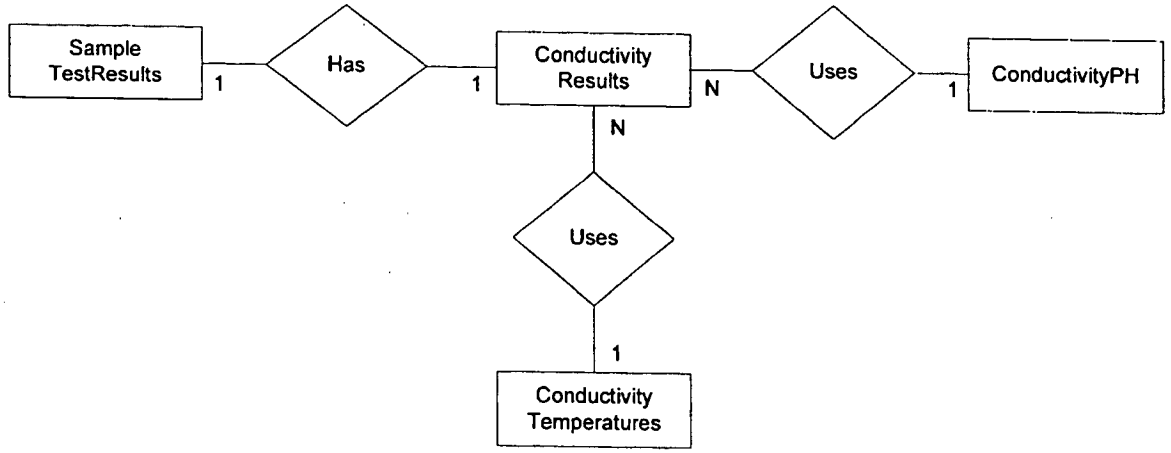

Incubators
1
Includes
N
Equipment
1
Includes
N
SampleTest Results
N
Has
1
SampleTestLevels
N
Includes
1
MediaLots

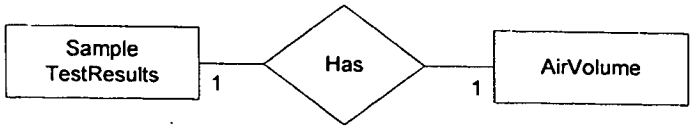

WorkLists
Has
SampleTest Results
Enters
Reviews
Belongs
Users
1
N
N
N
N
1
1
1

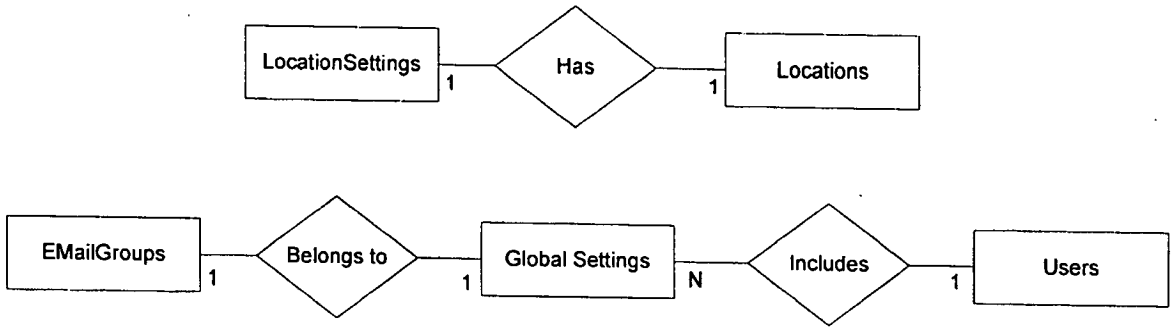

Genus
1
Has
N
Species
1
Has
N
Subspecies

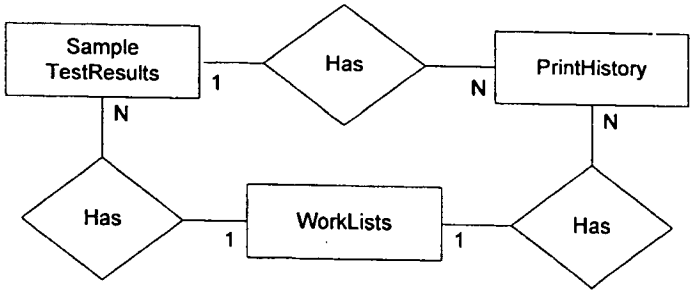

Morphology Headings
1
Describes
N
Morphology Descriptions
N
Has
N
OrganismMorphology Descriptions
N
Has
N
Organisms
N
Identifies
1
Users
Reviews
N
1
N
1
Has
N
Sample TestResults
N
Enters

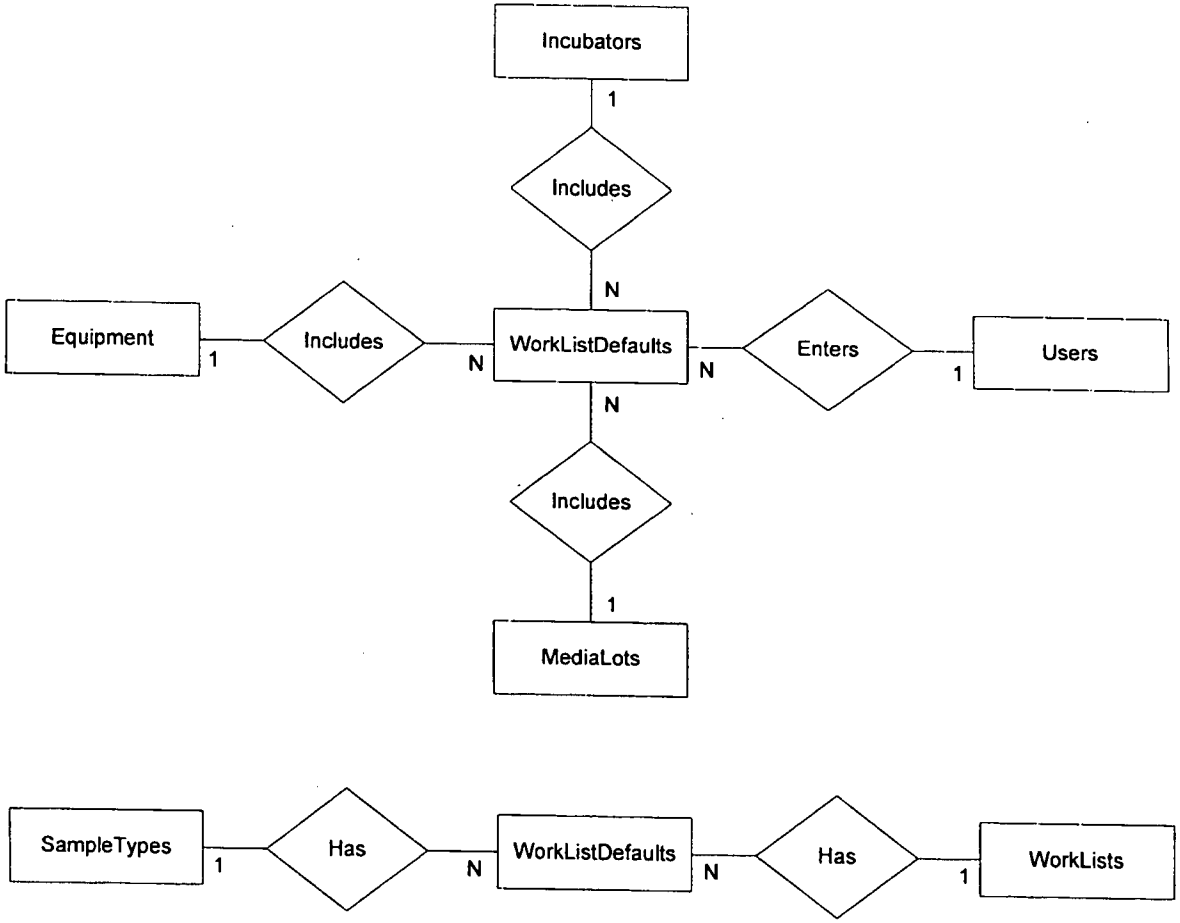

Sample TestResults
1
Has
1
Exceptions
1
Has
1
ExceptionAction History

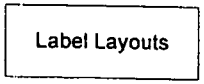

Sample TestResults
1
Has
1
Conductivity Results
N
Uses
1
ConductivityPH
N
Uses
1
Conductivity Temperatures


Sample
TestResults
1
Has
1
AirVolume


LocationSettings
1
Has
1
Locations
EMailGroups
1
Belongs to
1
Global Settings
N
Includes
1
Users


Sample
TestResults
1
Has
N
PrintHistory
N
N
Has
1
WorkLists
1
Has


Incubators
1
Includes
N
Equipment
1
Includes
N
WorkListDefaults
N
Enters
1
Users
N
Includes
1
MediaLots
SampleTypes
1
Has
N
WorkListDefaults
N
Has
1
WorkLists


Label Layouts

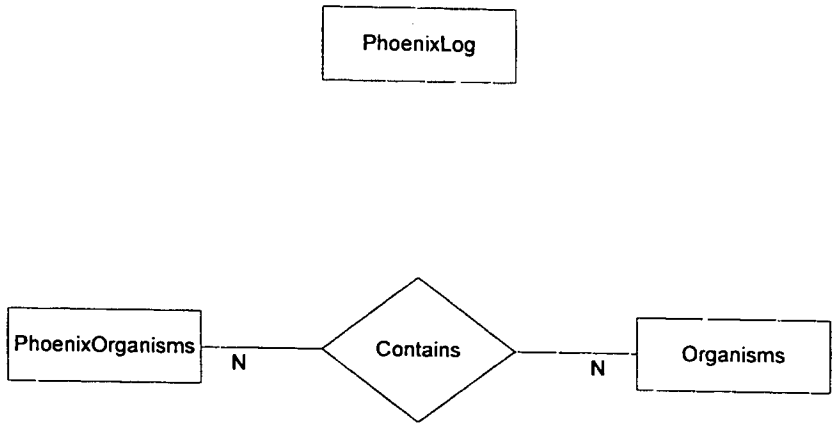

PhoenixLog
PhoenixOrganisms
N
Contains
N
Organisms

EXHIBIT G

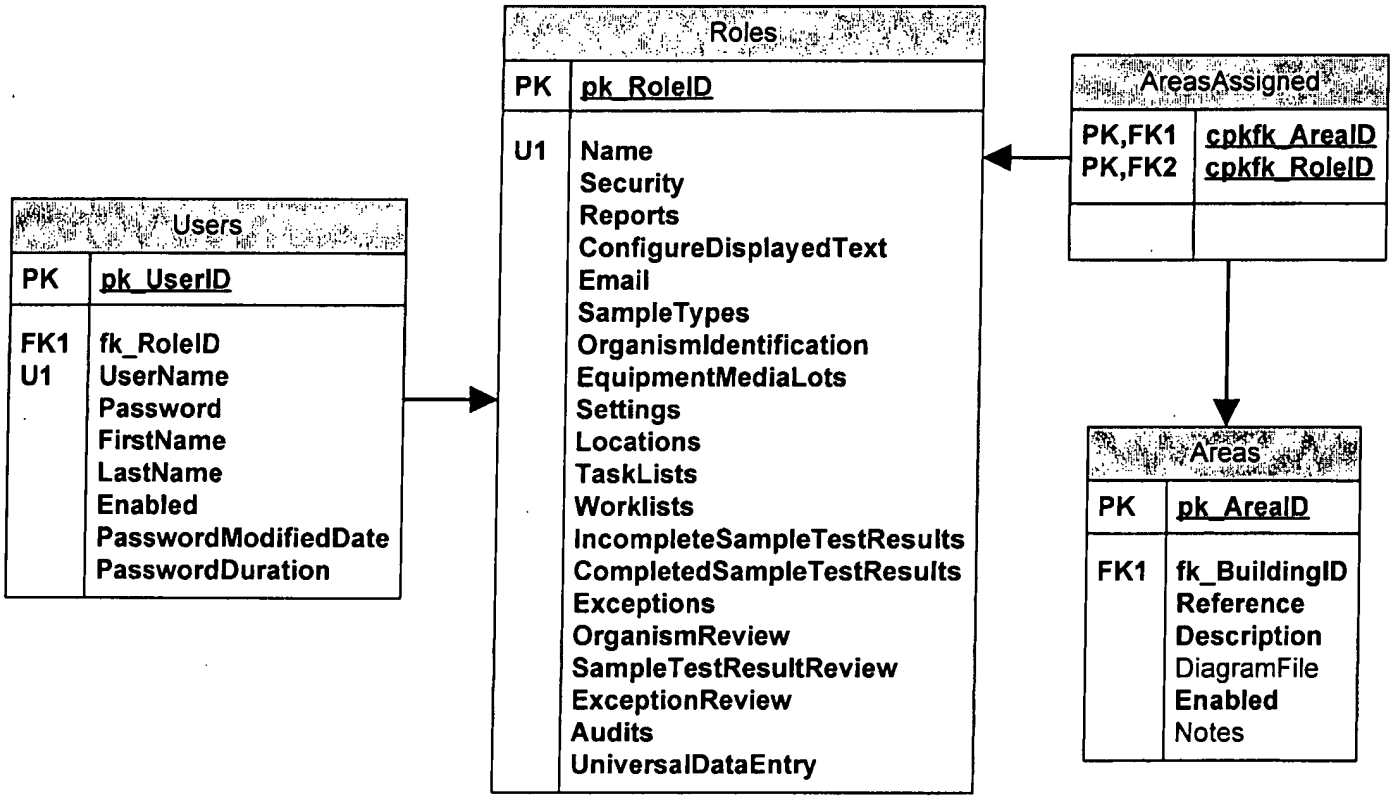
Product: EMS v. 3.0		Document: Table Definition ERD
Created By: Cynthia Jones		Revision By: Tony Castellano
Version: 22	Date: 05/09/03	Attachment A

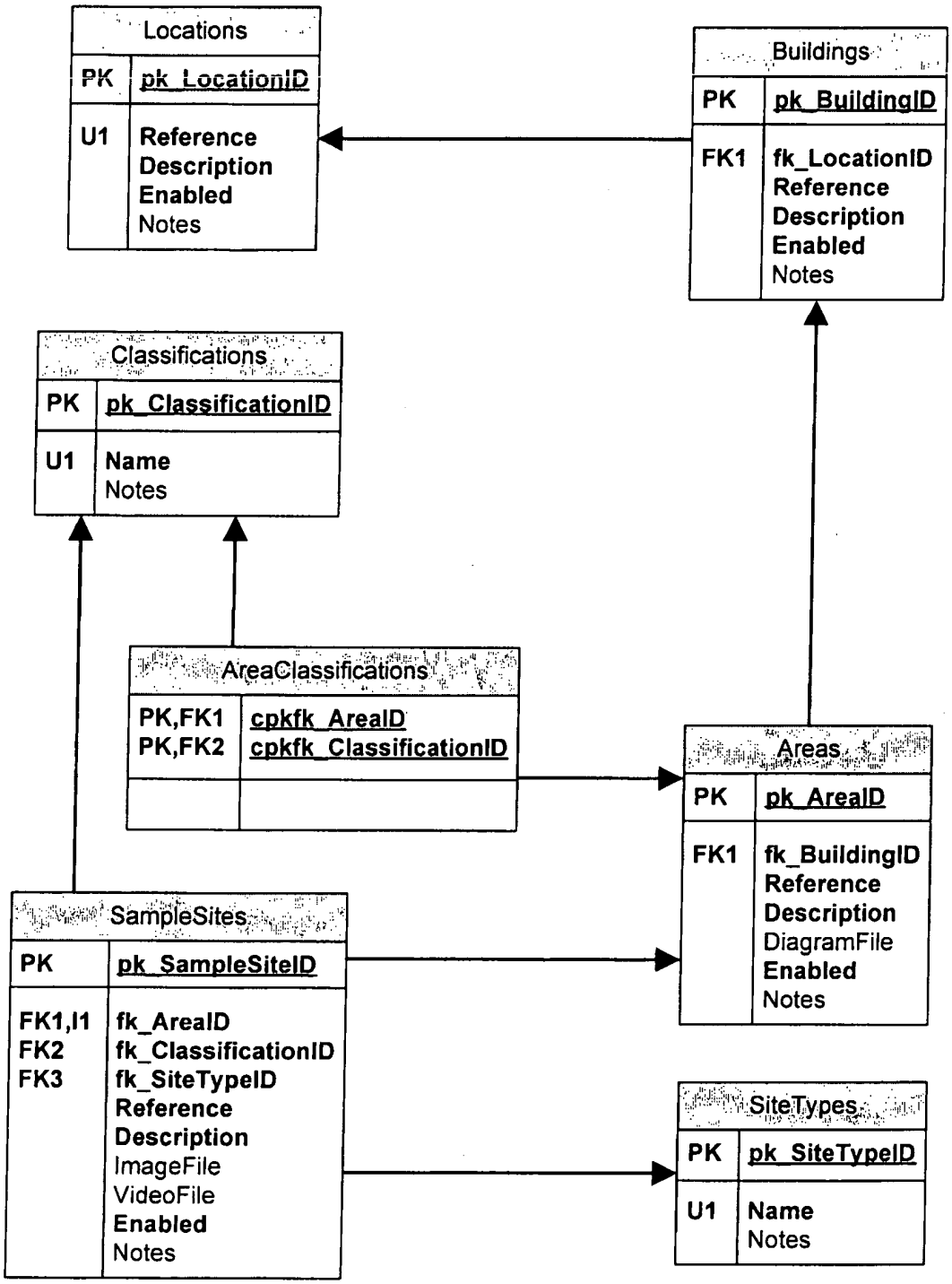
Revision Summary

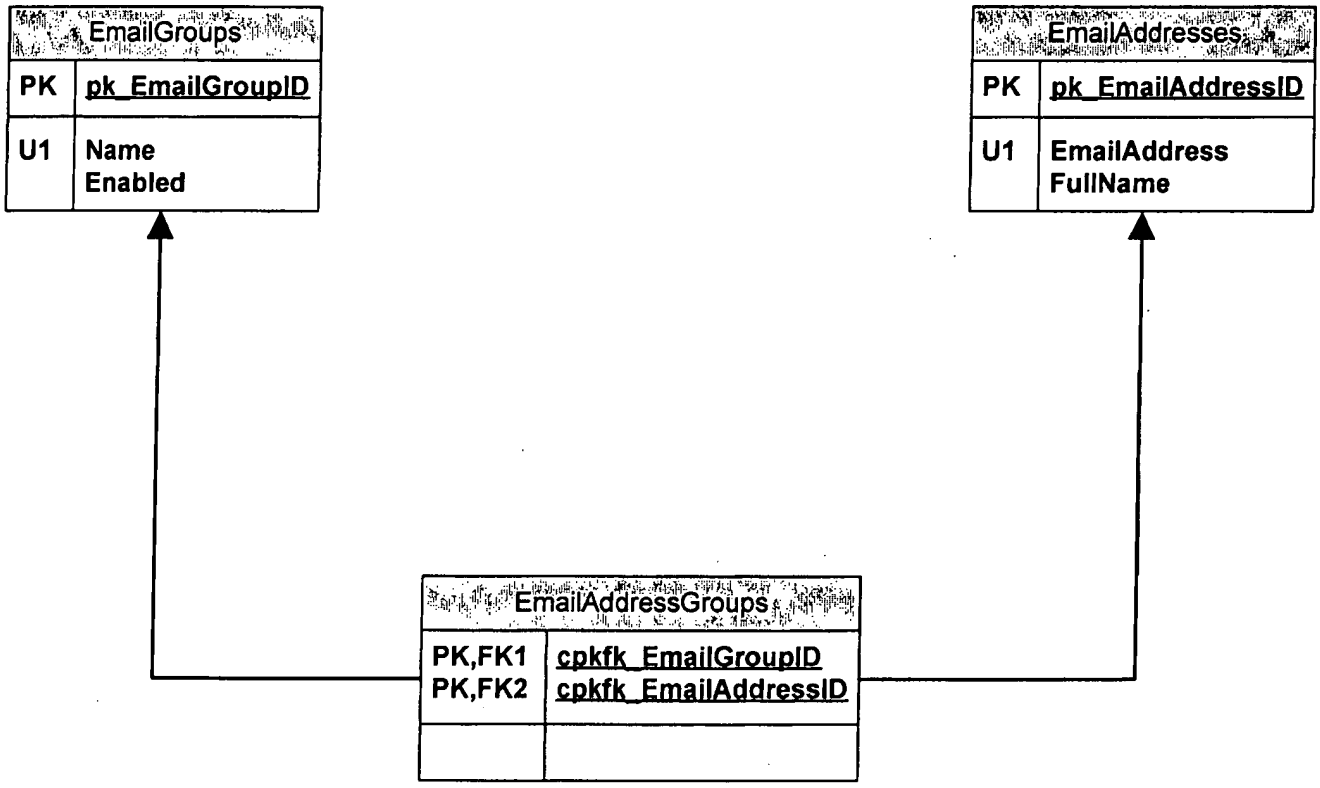
Added the 'PhoenixLog' table

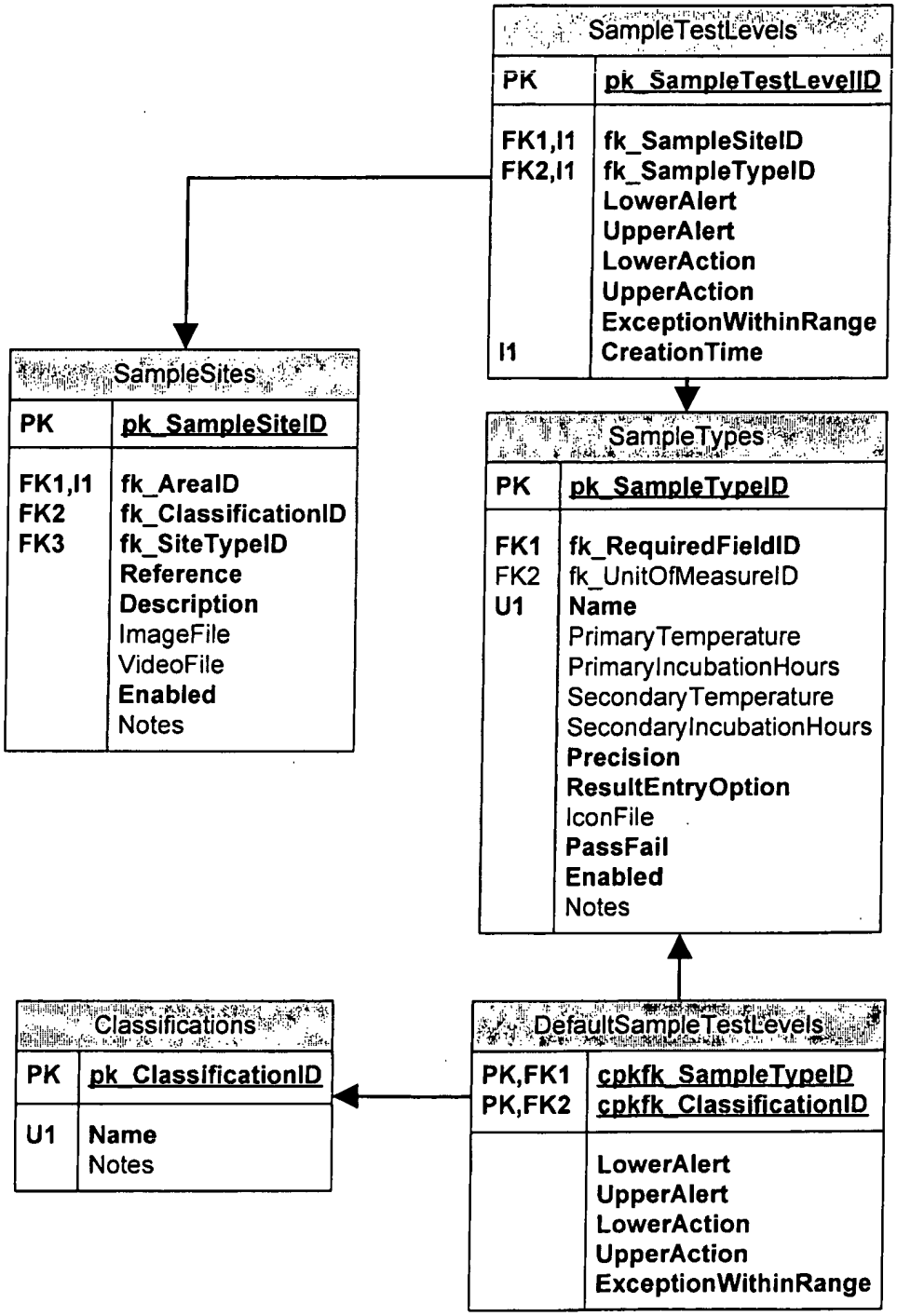
Added the 'PhoenixOrganisms' table

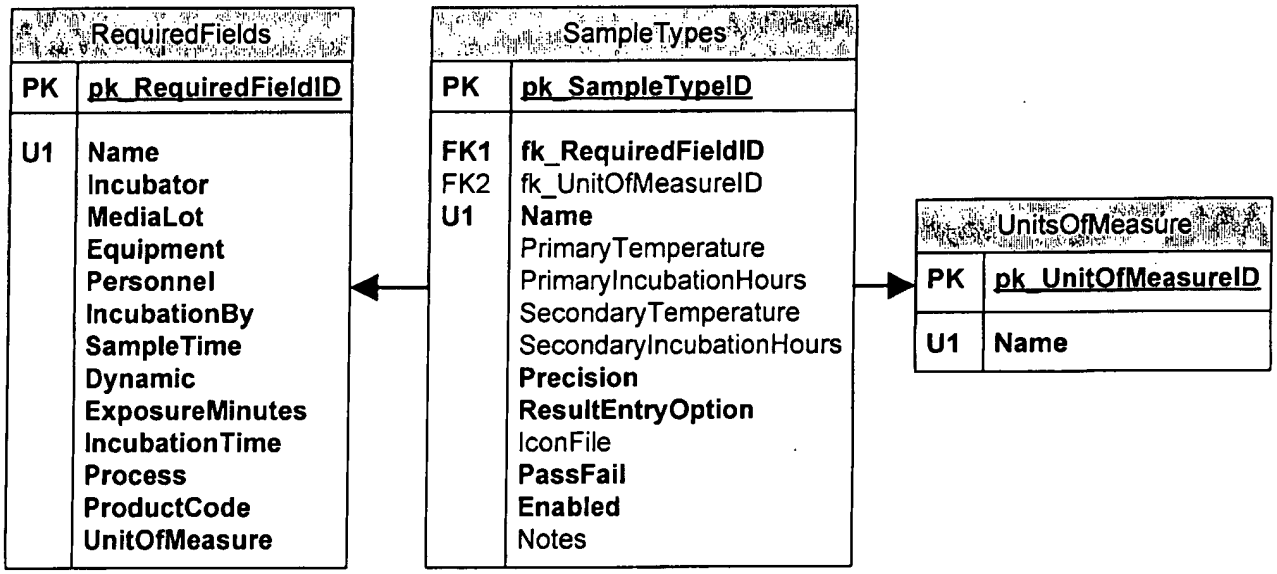
Added 'IsManualEntry' field to the 'SampleTestResultOrganisms' table

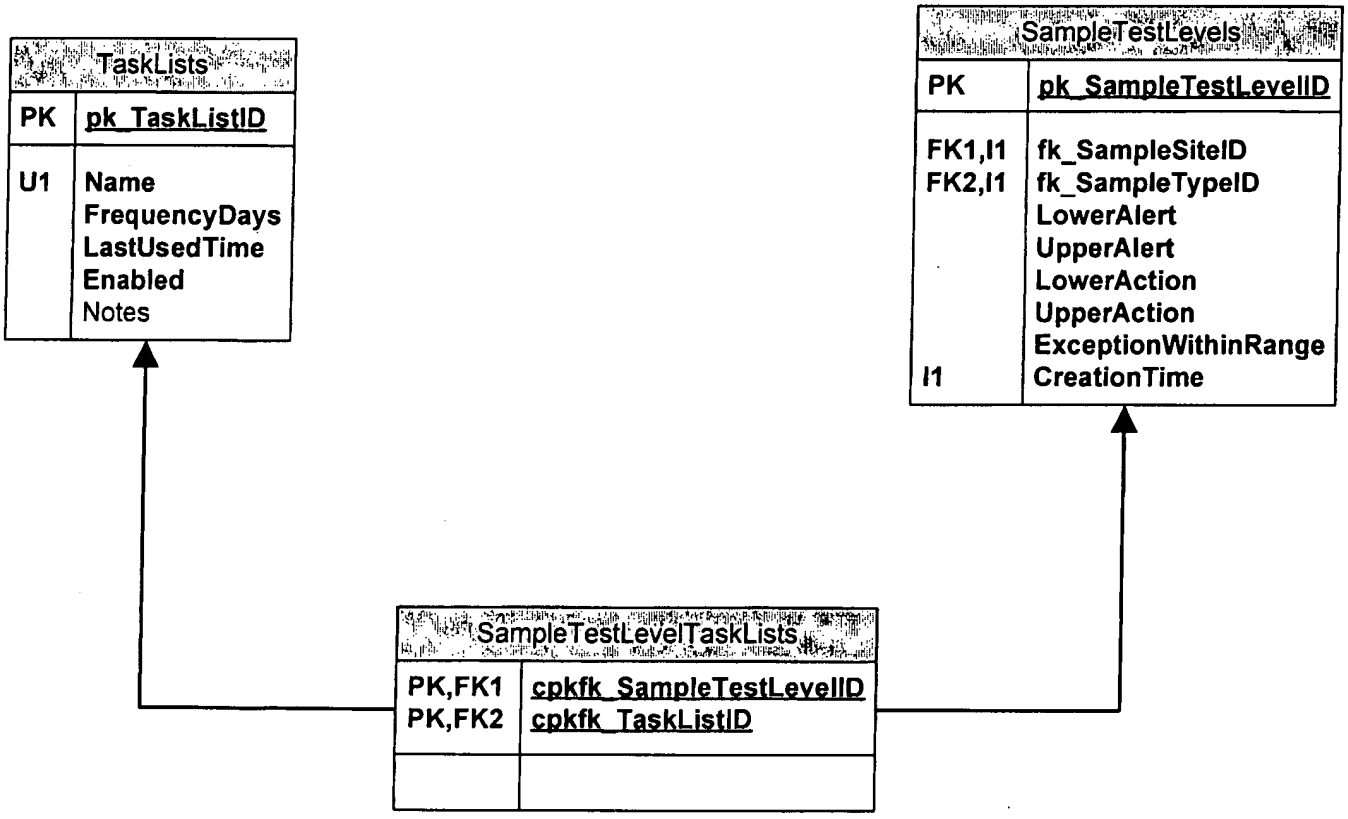

Roles
PK
pk_RoleID
U1
Name
Security
Reports
ConfigureDisplayedText
Email
SampleTypes
OrganismIdentification
EquipmentMediaLots
Settings
Locations
TaskLists
Worklists
IncompleteSampleTestResults
CompletedSampleTestResults
Exceptions
OrganismReview
SampleTestResultReview
ExceptionReview
Audits
UniversalDataEntry
AreasAssigned
PK,FK1
PK,FK2
cpkfk_AreaID
cpkfk_RoleID
Users
PK
pk_UserID
FK1
U1
fk_RoleID
UserName
Password
FirstName
LastName
Enabled
PasswordModifiedDate
PasswordDuration
Areas
PK
pk_AreaID
FK1
fk_BuildingID
Reference
Description
DiagramFile
Enabled
Notes

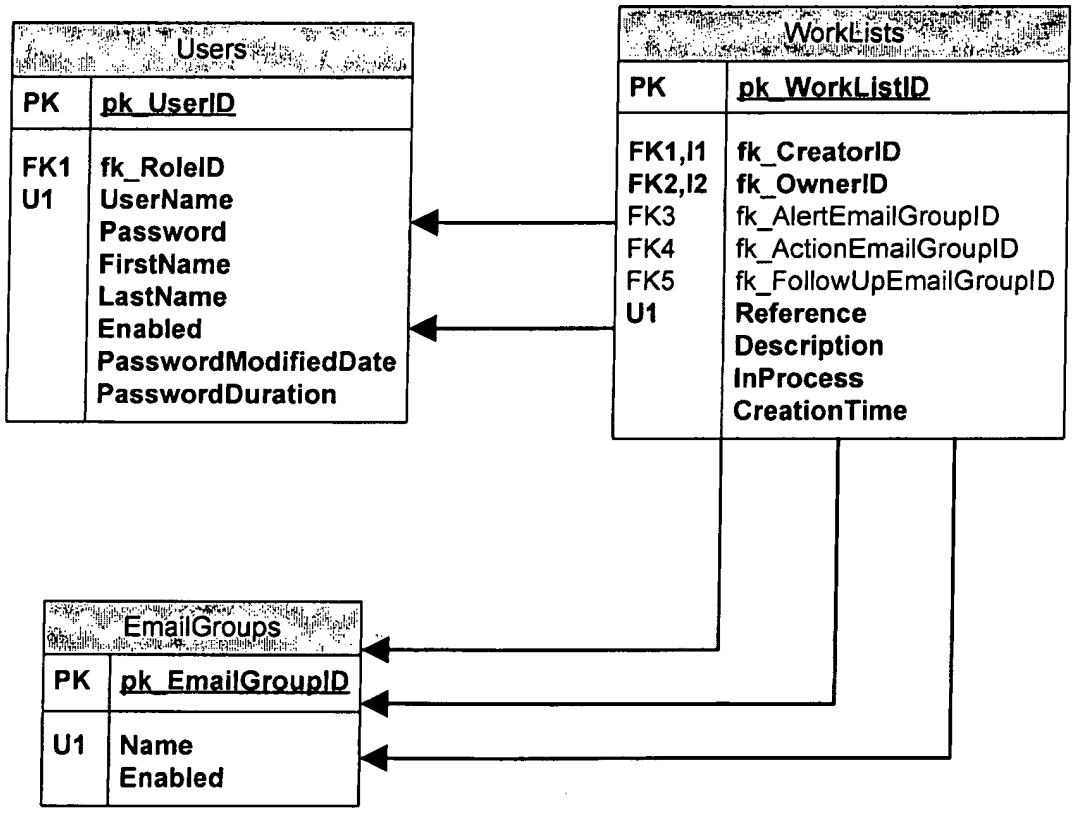

Locations
PK pk_LocationID
U1 Reference
Description
Enabled
Notes
Buildings
PK pk_BuildingID
FK1 fk_LocationID
Reference
Description
Enabled
Notes
Classifications
PK pk_ClassificationID
U1 Name
Notes
AreaClassifications
PK,FK1 cpkfk_AreaID
PK,FK2 cpkfk_ClassificationID
Areas
PK pk_AreaID
FK1 fk_BuildingID
Reference
Description
DiagramFile
Enabled
Notes
SampleSites
PK pk_SampleSiteID
FK1,I1 fk_AreaID
FK2 fk_ClassificationID
FK3 fk_SiteTypeID
Reference
Description
ImageFile
VideoFile
Enabled
Notes
SiteTypes
PK pk_SiteTypeID
U1 Name
Notes

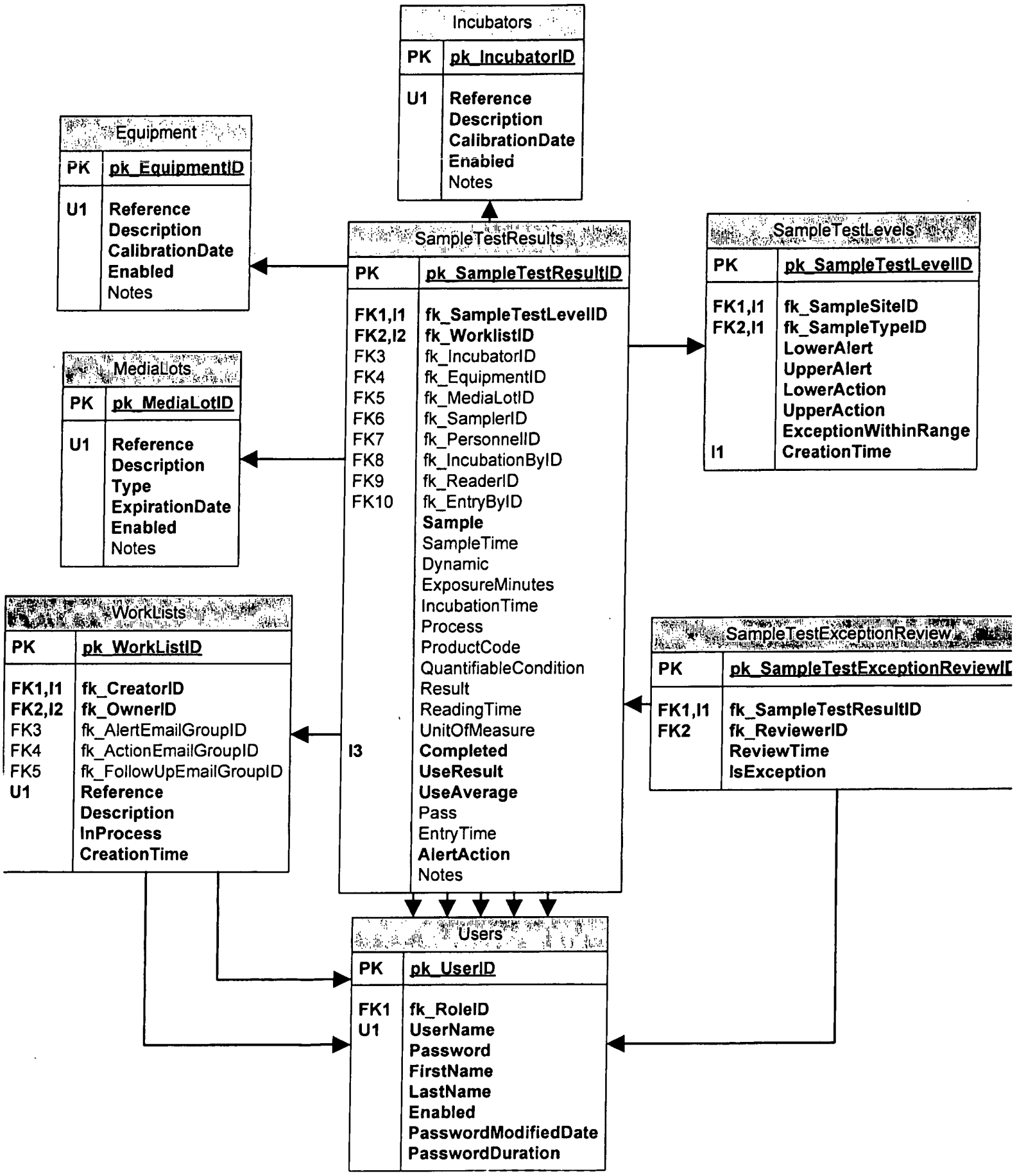

EmailGroups
PK
pk_EmailGroupID
U1
Name
Enabled
EmailAddresses
PK
pk_EmailAddressID
U1
EmailAddress
FullName
EmailAddressGroups
PK,FK1
PK,FK2
cpkfk_EmailGroupID
cpkfk_EmailAddressID

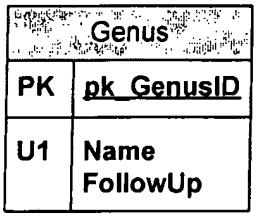

SampleTestLevels
PK
pk_SampleTestLevelID
FK1,I1
FK2,I1
I1
fk_SampleSiteID
fk_SampleTypeID
LowerAlert
UpperAlert
LowerAction
UpperAction
ExceptionWithinRange
CreationTime
SampleSites
PK
pk_SampleSiteID
FK1,I1
FK2
FK3
fk_AreaID
fk_ClassificationID
fk_SiteTypeID
Reference
Description
ImageFile
VideoFile
Enabled
Notes
SampleTypes
PK
pk_SampleTypeID
FK1
FK2
U1
fk_RequiredFieldID
fk_UnitOfMeasureID
Name
PrimaryTemperature
PrimaryIncubationHours
SecondaryTemperature
SecondaryIncubationHours
Precision
ResultEntryOption
IconFile
PassFail
Enabled
Notes
Classifications
PK
pk_ClassificationID
U1
Name
Notes
DefaultSampleTestLevels
PK,FK1
PK,FK2
cpkfk_SampleTypeID
cpkfk_ClassificationID
LowerAlert
UpperAlert
LowerAction
UpperAction
ExceptionWithinRange

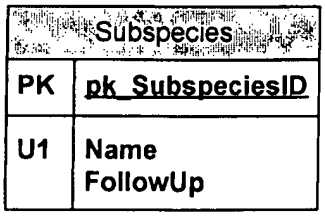

RequiredFields
PK
pk_RequiredFieldID
U1
Name
Incubator
MediaLot
Equipment
Personnel
IncubationBy
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
UnitOfMeasure
SampleTypes
PK
pk_SampleTypeID
FK1
fk_RequiredFieldID
FK2
fk_UnitOfMeasureID
U1
Name
PrimaryTemperature
PrimaryIncubationHours
SecondaryTemperature
SecondaryIncubationHours
Precision
ResultEntryOption
IconFile
PassFail
Enabled
Notes
UnitsOfMeasure
PK
pk_UnitOfMeasureID
U1
Name

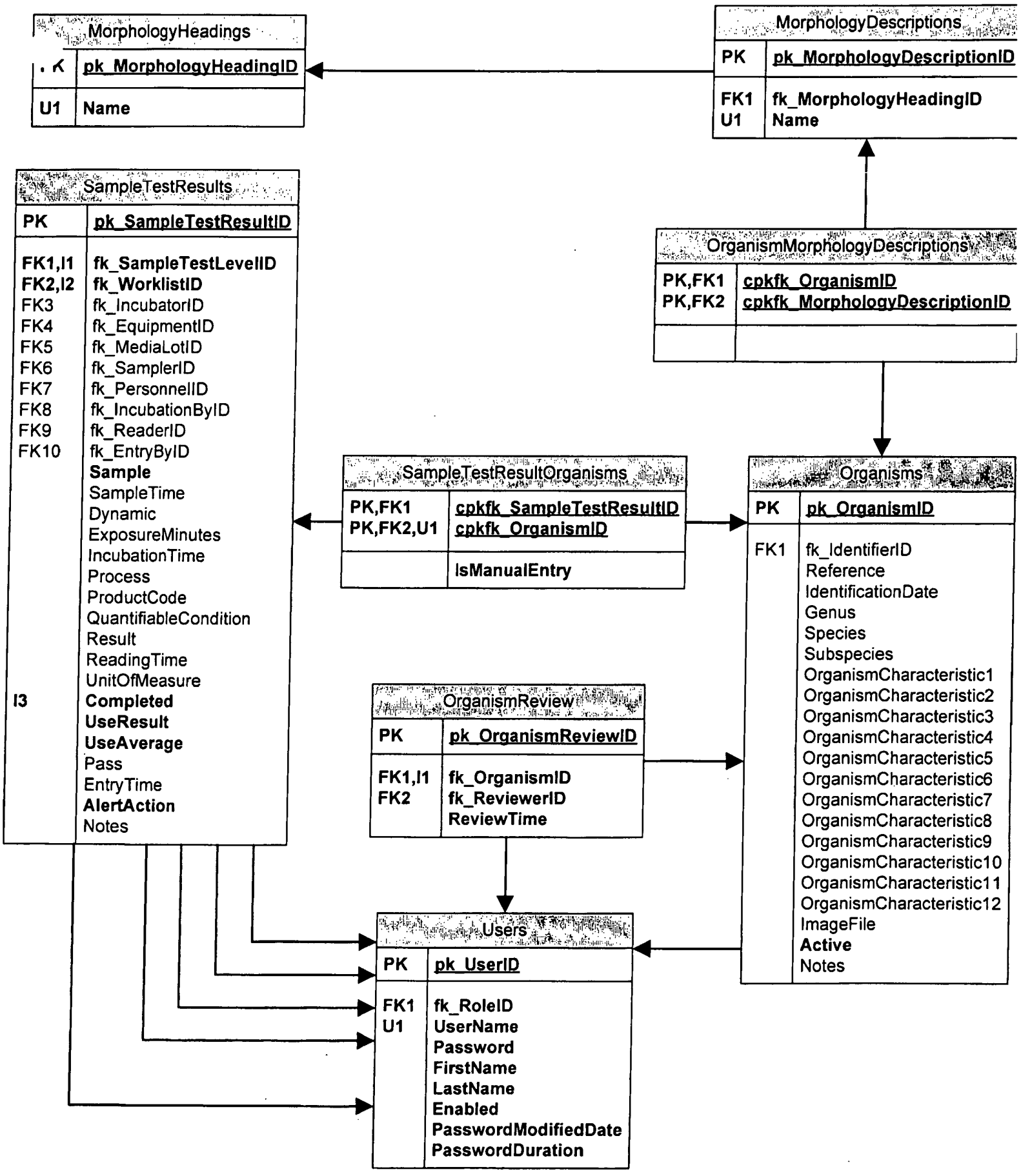

TaskLists
PK
pk_TaskListID
U1
Name
FrequencyDays
LastUsedTime
Enabled
Notes
SampleTestLevels
PK
pk_SampleTestLevelID
FK1,I1
FK2,I1
fk_SampleSiteID
fk_SampleTypeID
LowerAlert
UpperAlert
LowerAction
UpperAction
ExceptionWithinRange
I1
CreationTime
SampleTestLevelTaskLists
PK,FK1
PK,FK2
cpkfk_SampleTestLevelID
cpkfk_TaskListID

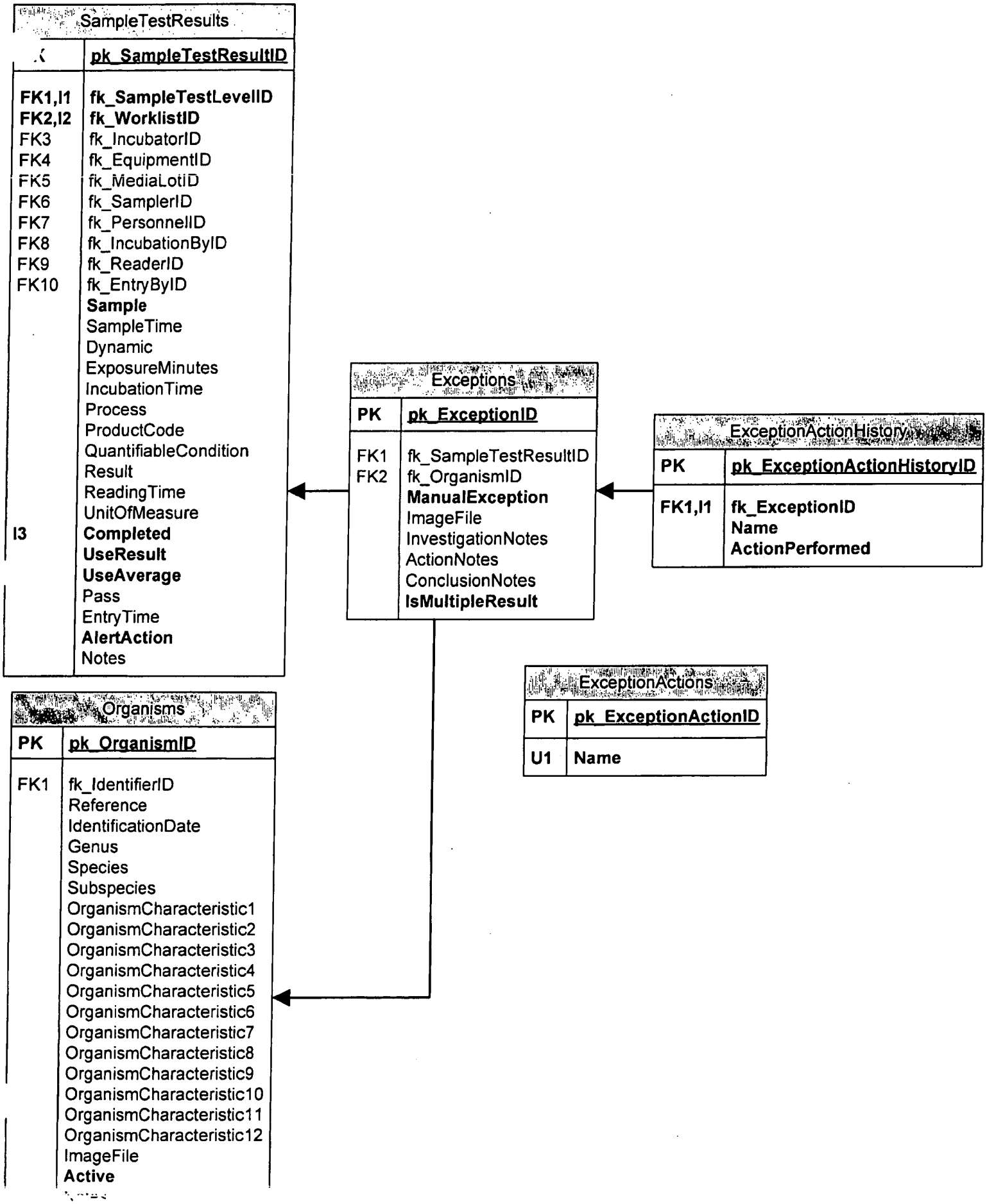

Users
PK
pk_UserID
FK1
U1
fk_RoleID
UserName
Password
FirstName
LastName
Enabled
PasswordModifiedDate
PasswordDuration
WorkLists
PK
pk_WorkListID
FK1,I1
FK2,I2
FK3
FK4
FK5
U1
fk_CreatorID
fk_OwnerID
fk_AlertEmailGroupID
fk_ActionEmailGroupID
fk_FollowUpEmailGroupID
Reference
Description
InProcess
CreationTime
EmailGroups
PK
pk_EmailGroupID
U1
Name
Enabled

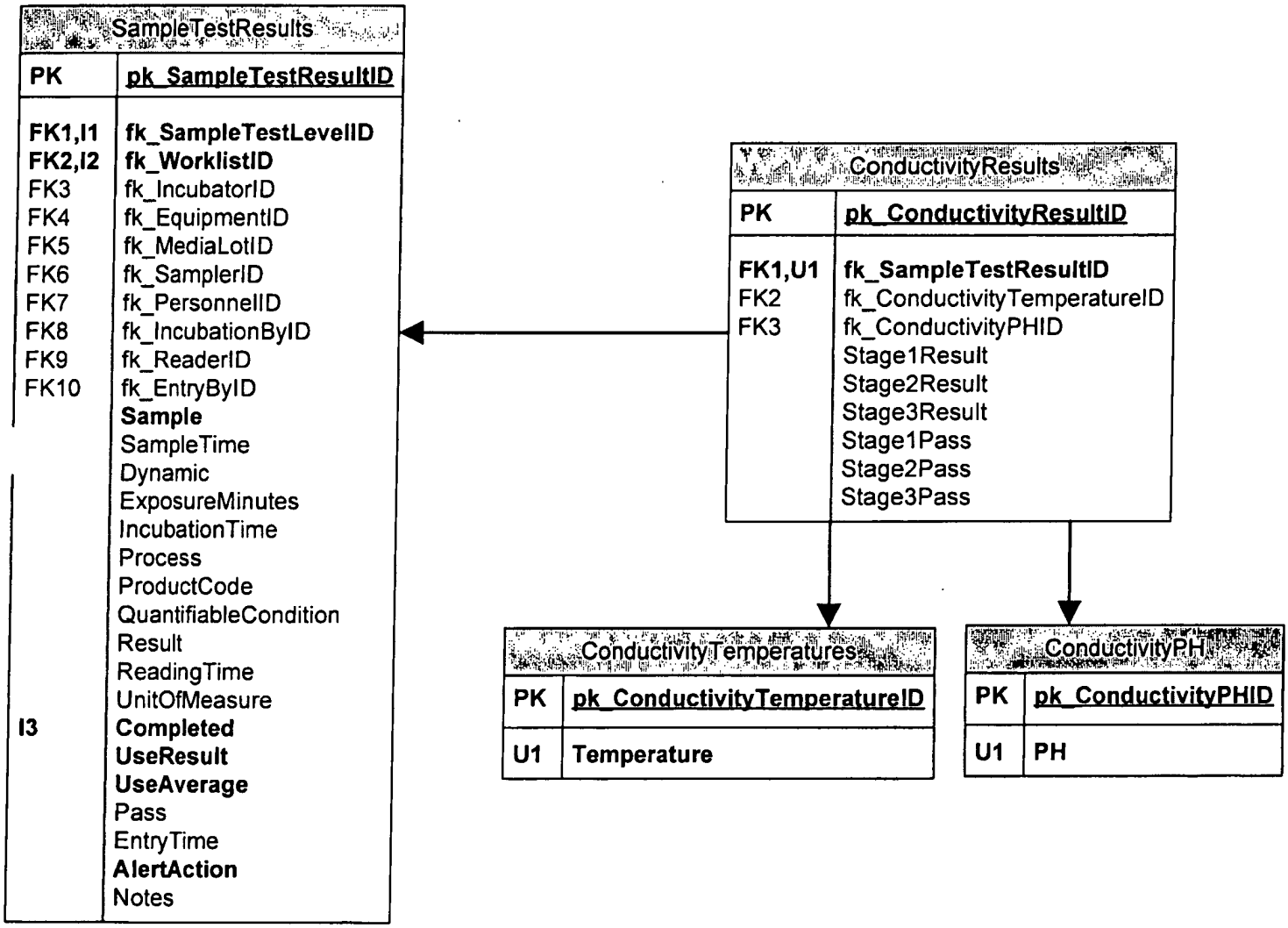

Incubators
PK pk_IncubatorID
U1 Reference
Description
CalibrationDate
Enabled
Notes
Equipment
PK pk_EquipmentID
U1 Reference
Description
CalibrationDate
Enabled
Notes
SampleTestResults
PK pk_SampleTestResultID
FK1,I1 fk_SampleTestLevelID
FK2,I2 fk_WorklistID
FK3 fk_IncubatorID
FK4 fk_EquipmentID
FK5 fk_MediaLotID
FK6 fk_SamplerID
FK7 fk_PersonnelID
FK8 fk_IncubationByID
FK9 fk_ReaderID
FK10 fk_EntryByID
Sample
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
QuantifiableCondition
Result
ReadingTime
UnitOfMeasure
I3 Completed
UseResult
UseAverage
Pass
EntryTime
AlertAction
Notes
SampleTestLevels
PK pk_SampleTestLevelID
FK1,I1 fk_SampleSiteID
FK2,I1 fk_SampleTypeID
LowerAlert
UpperAlert
LowerAction
UpperAction
ExceptionWithinRange
I1 CreationTime
MediaLots
PK pk_MediaLotID
U1 Reference
Description
Type
ExpirationDate
Enabled
Notes
WorkLists
PK pk_WorkListID
FK1,I1 fk_CreatorID
FK2,I2 fk_OwnerID
FK3 fk_AlertEmailGroupID
FK4 fk_ActionEmailGroupID
FK5 fk_FollowUpEmailGroupID
U1 Reference
Description
InProcess
CreationTime
SampleTestExceptionReview
PK pk_SampleTestExceptionReviewID
FK1,I1 fk_SampleTestResultID
FK2 fk_ReviewerID
ReviewTime
IsException
Users
PK pk_UserID
FK1 fk_RoleID
U1 UserName
Password
FirstName
LastName
Enabled
PasswordModifiedDate
PasswordDuration

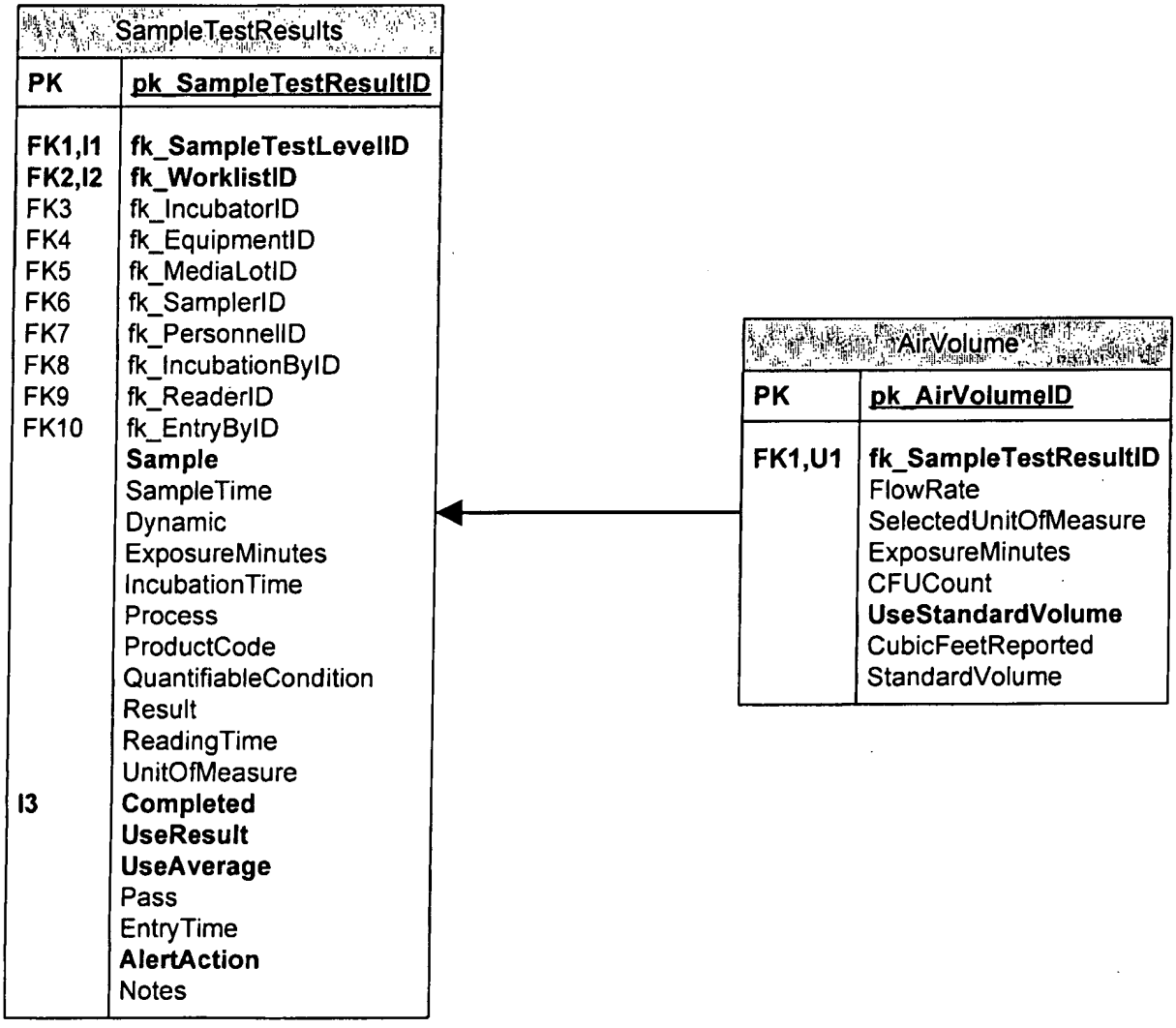

Genus
PK
pk_GenusID
U1
Name
FollowUp

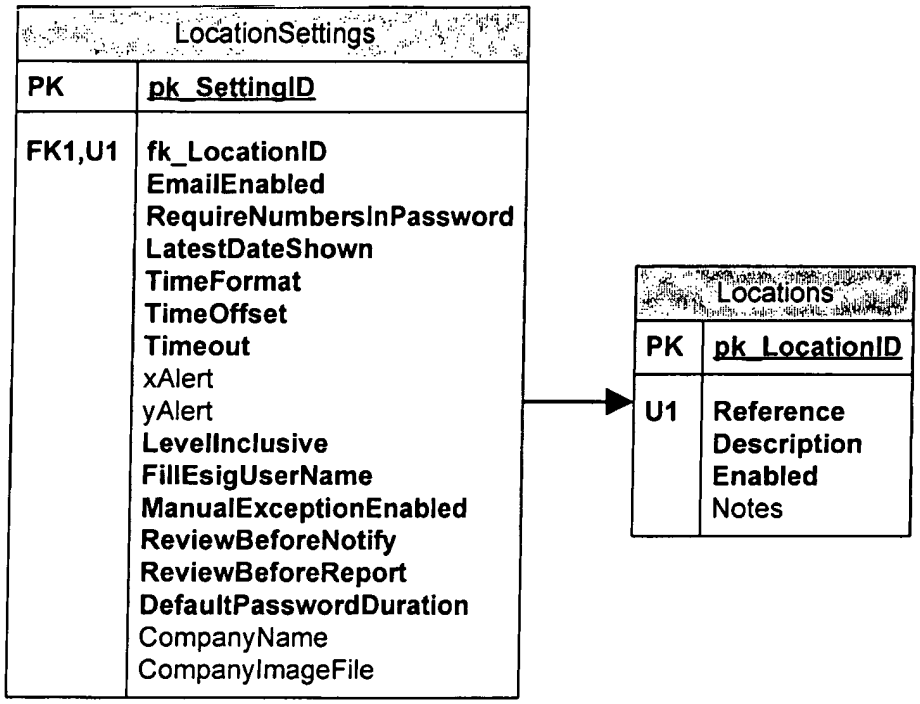

Subspecies
PK
pk_SubspeciesID
U1
Name
FollowUp

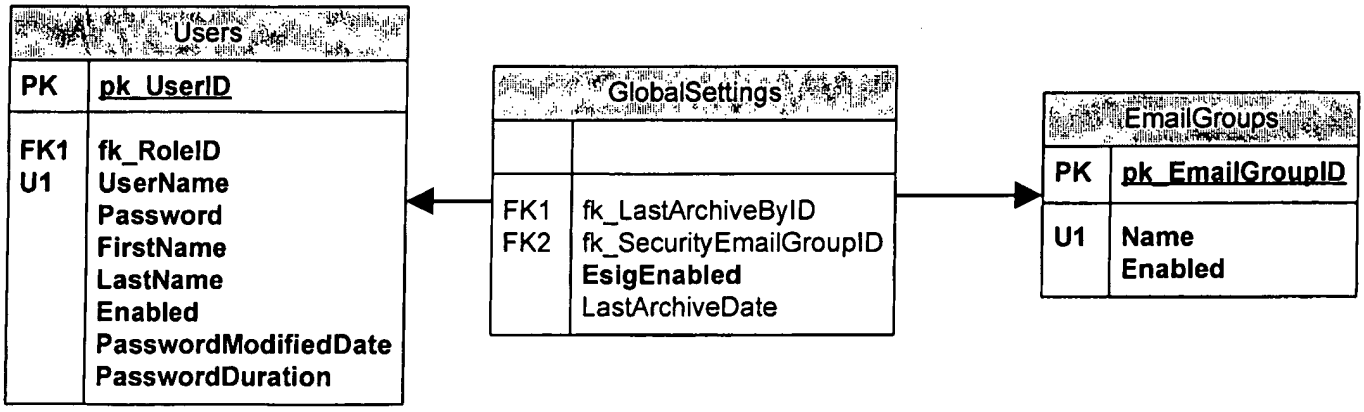

MorphologyHeadings
pk_MorphologyHeadingID
U1 Name
MorphologyDescriptions
PK pk_MorphologyDescriptionID
FK1 fk_MorphologyHeadingID
U1 Name
SampleTestResults
PK pk_SampleTestResultID
FK1,I1 fk_SampleTestLevelID
FK2,I2 fk_WorklistID
FK3 fk_IncubatorID
FK4 fk_EquipmentID
FK5 fk_MediaLotID
FK6 fk_SamplerID
FK7 fk_PersonnelID
FK8 fk_IncubationByID
FK9 fk_ReaderID
FK10 fk_EntryByID
Sample
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
QuantifiableCondition
Result
ReadingTime
UnitOfMeasure
I3 Completed
UseResult
UseAverage
Pass
EntryTime
AlertAction
Notes
OrganismMorphologyDescriptions
PK,FK1 cpkfk_OrganismID
PK,FK2 cpkfk_MorphologyDescriptionID
SampleTestResultOrganisms
PK,FK1 cpkfk_SampleTestResultID
PK,FK2,U1 cpkfk_OrganismID
IsManualEntry
Organisms
PK pk_OrganismID
FK1 fk_IdentifierID
Reference
IdentificationDate
Genus
Species
Subspecies
OrganismCharacteristic1
OrganismCharacteristic2
OrganismCharacteristic3
OrganismCharacteristic4
OrganismCharacteristic5
OrganismCharacteristic6
OrganismCharacteristic7
OrganismCharacteristic8
OrganismCharacteristic9
OrganismCharacteristic10
OrganismCharacteristic11
OrganismCharacteristic12
ImageFile
Active
Notes
OrganismReview
PK pk_OrganismReviewID
FK1,I1 fk_OrganismID
FK2 fk_ReviewerID
ReviewTime
Users
PK pk_UserID
FK1 fk_RoleID
U1 UserName
Password
FirstName
LastName
Enabled
PasswordModifiedDate
PasswordDuration

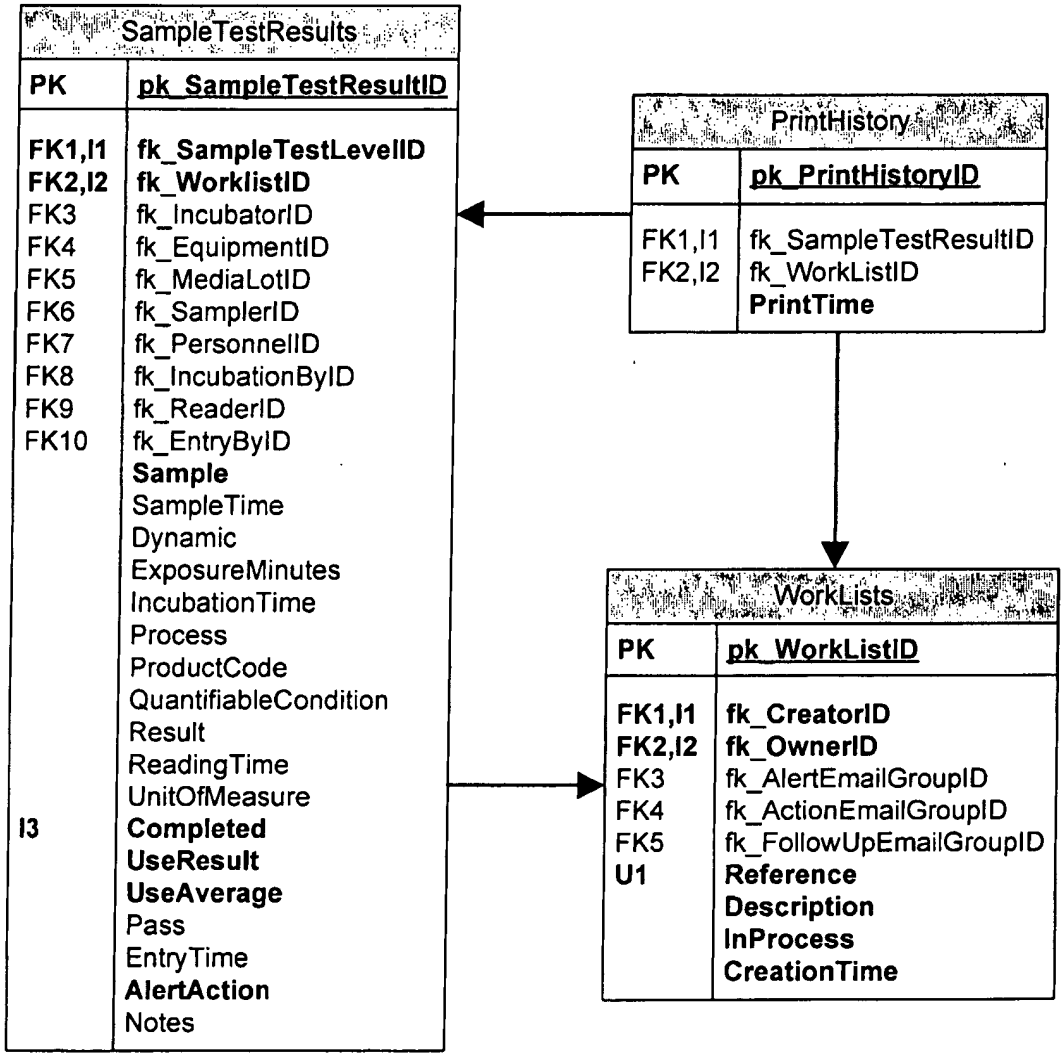

SampleTestResults
pk_SampleTestResultID
FK1,I1 fk_SampleTestLevelID
FK2,I2 fk_WorklistID
FK3 fk_IncubatorID
FK4 fk_EquipmentID
FK5 fk_MediaLotID
FK6 fk_SamplerID
FK7 fk_PersonnelID
FK8 fk_IncubationByID
FK9 fk_ReaderID
FK10 fk_EntryByID
Sample
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
QuantifiableCondition
Result
ReadingTime
UnitOfMeasure
I3 Completed
UseResult
UseAverage
Pass
EntryTime
AlertAction
Notes
Exceptions
PK pk_ExceptionID
FK1 fk_SampleTestResultID
FK2 fk_OrganismID
ManualException
ImageFile
InvestigationNotes
ActionNotes
ConclusionNotes
IsMultipleResult
ExceptionActionHistory
PK pk_ExceptionActionHistoryID
FK1,I1 fk_ExceptionID
Name
ActionPerformed
ExceptionActions
PK pk_ExceptionActionID
U1 Name
Organisms
PK pk_OrganismID
FK1 fk_IdentifierID
Reference
IdentificationDate
Genus
Species
Subspecies
OrganismCharacteristic1
OrganismCharacteristic2
OrganismCharacteristic3
OrganismCharacteristic4
OrganismCharacteristic5
OrganismCharacteristic6
OrganismCharacteristic7
OrganismCharacteristic8
OrganismCharacteristic9
OrganismCharacteristic10
OrganismCharacteristic11
OrganismCharacteristic12
ImageFile
Active

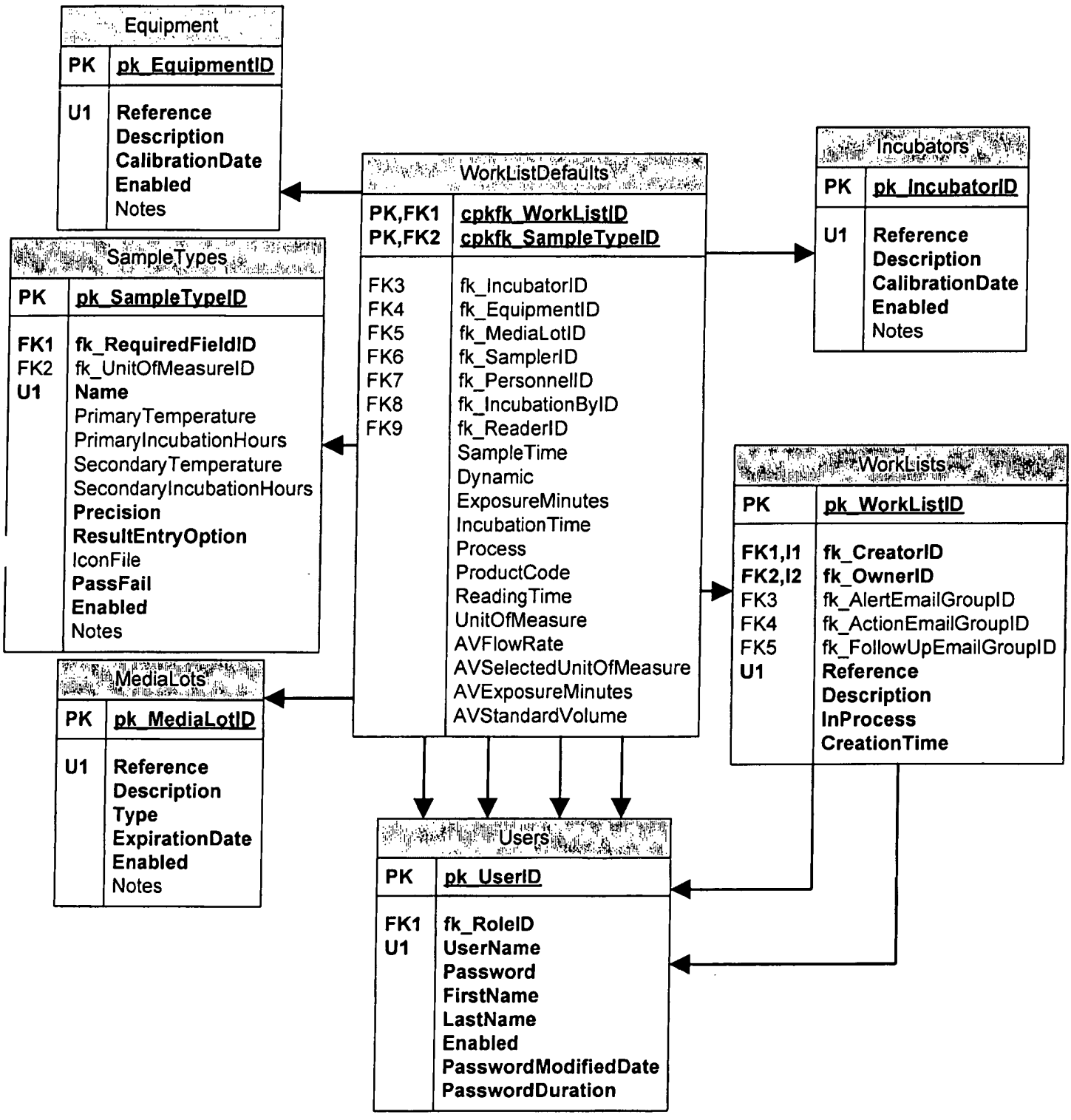

SampleTestResults
PK
pk_SampleTestResultID
FK1,I1
fk_SampleTestLevelID
FK2,I2
fk_WorklistID
FK3
fk_IncubatorID
FK4
fk_EquipmentID
FK5
fk_MediaLotID
FK6
fk_SamplerID
FK7
fk_PersonnelID
FK8
fk_IncubationByID
FK9
fk_ReaderID
FK10
fk_EntryByID
Sample
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
QuantifiableCondition
Result
ReadingTime
UnitOfMeasure
I3
Completed
UseResult
UseAverage
Pass
EntryTime
AlertAction
Notes
ConductivityResults
PK
pk_ConductivityResultID
FK1,U1
fk_SampleTestResultID
FK2
fk_ConductivityTemperatureID
FK3
fk_ConductivityPHID
Stage1Result
Stage2Result
Stage3Result
Stage1Pass
Stage2Pass
Stage3Pass
ConductivityTemperatures
PK
pk_ConductivityTemperatureID
U1
Temperature
ConductivityPH
PK
pk_ConductivityPHID
U1
PH

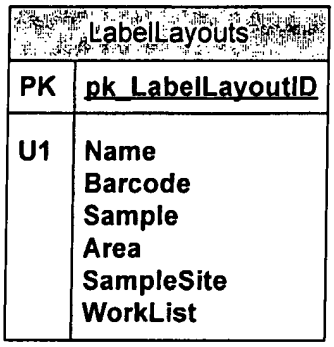

SampleTestResults
PK
pk_SampleTestResultID
FK1,I1
fk_SampleTestLevelID
FK2,I2
fk_WorklistID
FK3
fk_IncubatorID
FK4
fk_EquipmentID
FK5
fk_MediaLotID
FK6
fk_SamplerID
FK7
fk_PersonnelID
FK8
fk_IncubationByID
FK9
fk_ReaderID
FK10
fk_EntryByID
Sample
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
QuantifiableCondition
Result
ReadingTime
UnitOfMeasure
I3
Completed
UseResult
UseAverage
Pass
EntryTime
AlertAction
Notes
AirVolume
PK
pk_AirVolumeID
FK1,U1
fk_SampleTestResultID
FlowRate
SelectedUnitOfMeasure
ExposureMinutes
CFUCount
UseStandardVolume
CubicFeetReported
StandardVolume

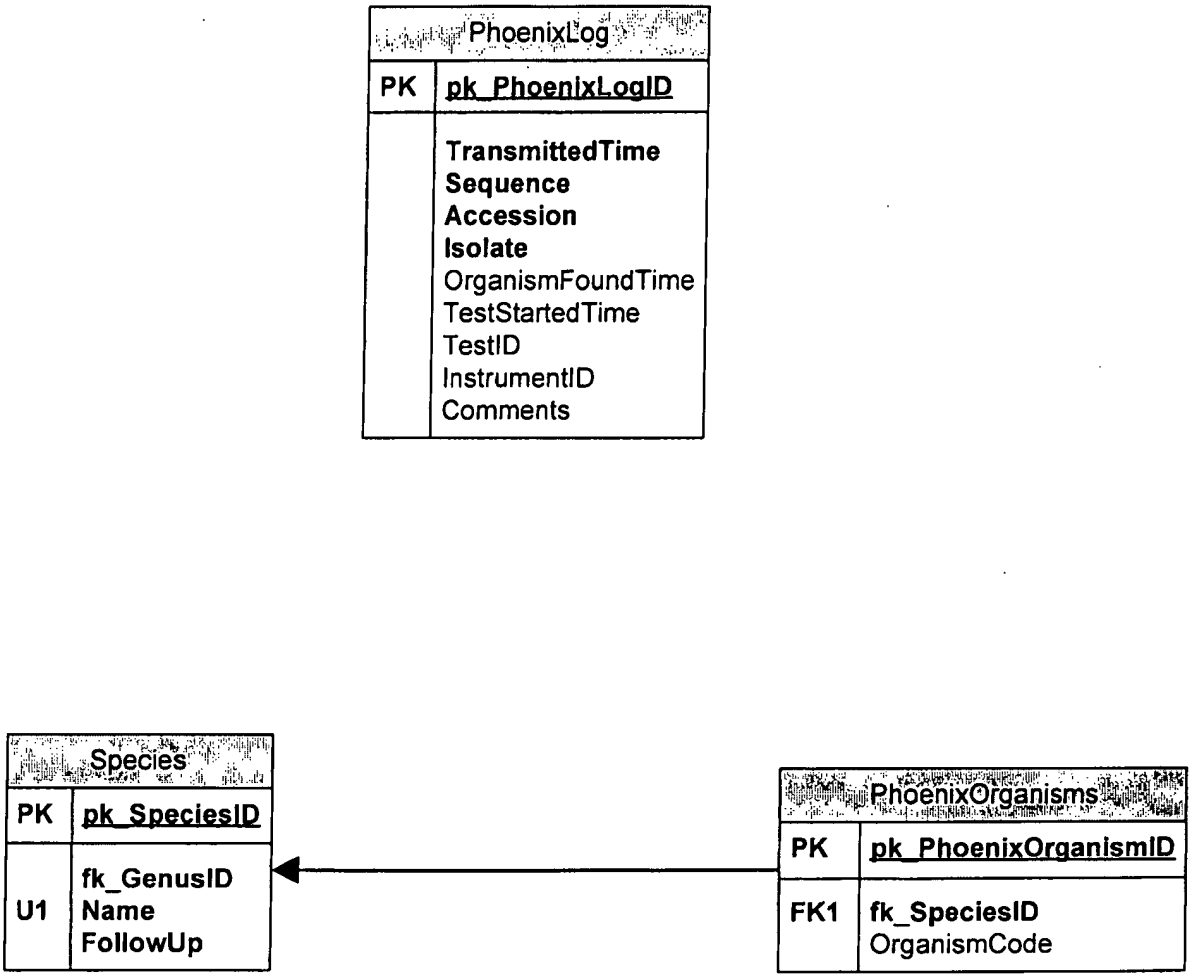

LocationSettings
PK
pk_SettingID
FK1,U1
fk_LocationID
EmailEnabled
RequireNumbersInPassword
LatestDateShown
TimeFormat
TimeOffset
Timeout
xAlert
yAlert
LevelInclusive
FillEsigUserName
ManualExceptionEnabled
ReviewBeforeNotify
ReviewBeforeReport
DefaultPasswordDuration
CompanyName
CompanyImageFile
Locations
PK
pk_LocationID
U1
Reference
Description
Enabled
Notes


Users
PK
pk_UserID
FK1
U1
fk_RoleID
UserName
Password
FirstName
LastName
Enabled
PasswordModifiedDate
PasswordDuration
GlobalSettings
FK1
FK2
fk_LastArchiveByID
fk_SecurityEmailGroupID
EsigEnabled
LastArchiveDate
EmailGroups
PK
pk_EmailGroupID
U1
Name
Enabled


SampleTestResults
PK
pk_SampleTestResultID
FK1,I1
FK2,I2
FK3
FK4
FK5
FK6
FK7
FK8
FK9
FK10
I3
fk_SampleTestLevelID
fk_WorklistID
fk_IncubatorID
fk_EquipmentID
fk_MediaLotID
fk_SamplerID
fk_PersonnelID
fk_IncubationByID
fk_ReaderID
fk_EntryByID
Sample
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
QuantifiableCondition
Result
ReadingTime
UnitOfMeasure
Completed
UseResult
UseAverage
Pass
EntryTime
AlertAction
Notes
PrintHistory
PK
pk_PrintHistoryID
FK1,I1
FK2,I2
fk_SampleTestResultID
fk_WorkListID
PrintTime
WorkLists
PK
pk_WorkListID
FK1,I1
FK2,I2
FK3
FK4
FK5
U1
fk_CreatorID
fk_OwnerID
fk_AlertEmailGroupID
fk_ActionEmailGroupID
fk_FollowUpEmailGroupID
Reference
Description
InProcess
CreationTime


Equipment
PK pk_EquipmentID
U1 Reference
Description
CalibrationDate
Enabled
Notes
WorkListDefaults
PK,FK1 cpkfk_WorkListID
PK,FK2 cpkfk_SampleTypeID
FK3 fk_IncubatorID
FK4 fk_EquipmentID
FK5 fk_MediaLotID
FK6 fk_SamplerID
FK7 fk_PersonnelID
FK8 fk_IncubationByID
FK9 fk_ReaderID
SampleTime
Dynamic
ExposureMinutes
IncubationTime
Process
ProductCode
ReadingTime
UnitOfMeasure
AVFlowRate
AVSelectedUnitOfMeasure
AVExposureMinutes
AVStandardVolume
Incubators
PK pk_IncubatorID
U1 Reference
Description
CalibrationDate
Enabled
Notes
SampleTypes
PK pk_SampleTypeID
FK1 fk_RequiredFieldID
FK2 fk_UnitOfMeasureID
U1 Name
PrimaryTemperature
PrimaryIncubationHours
SecondaryTemperature
SecondaryIncubationHours
Precision
ResultEntryOption
IconFile
PassFail
Enabled
Notes
WorkLists
PK pk_WorkListID
FK1,I1 fk_CreatorID
FK2,I2 fk_OwnerID
FK3 fk_AlertEmailGroupID
FK4 fk_ActionEmailGroupID
FK5 fk_FollowUpEmailGroupID
U1 Reference
Description
InProcess
CreationTime
MediaLots
PK pk_MediaLotID
U1 Reference
Description
Type
ExpirationDate
Enabled
Notes
Users
PK pk_UserID
FK1 fk_RoleID
U1 UserName
Password
FirstName
LastName
Enabled
PasswordModifiedDate
PasswordDuration


LabelLayouts
PK
pk_LabelLayoutID
U1
Name
Barcode
Sample
Area
SampleSite
WorkList


PhoenixLog
PK
pk_PhoenixLogID
TransmittedTime
Sequence
Accession
Isolate
OrganismFoundTime
TestStartedTime
TestID
InstrumentID
Comments
Species
PK
pk_SpeciesID
U1
fk_GenusID
Name
FollowUp
PhoenixOrganisms
PK
pk_PhoenixOrganismID
FK1
fk_SpeciesID
OrganismCode